



Titre: Génération de maillage de peau pour géométries complexes par
Title: décomposition spatiale hiérarchique

Auteur: Aboubacar Sidiki Sambou Bagayogo
Author:

Date: 2001

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Bagayogo, A. S. S. (2001). Génération de maillage de peau pour géométries
Citation: complexes par décomposition spatiale hiérarchique [Ph.D. thesis, École
Polytechnique de Montréal]. PolyPublie. <https://publications.polymtl.ca/8709/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie:
PolyPublie URL: <https://publications.polymtl.ca/8709/>

**Directeurs de
recherche:**
Advisors:

Programme: Unspecified
Program:

UNIVERSITÉ DE MONTRÉAL

GÉNÉRATION DE MAILLAGE DE PEAU POUR
GÉOMÉTRIES COMPLEXES PAR DÉCOMPOSITION
SPATIALE HIÉRARCHIQUE

ABOUBACAR SIDIKI SAMBOU BAGAYOGO
DÉPARTEMENT DE MATHÉMATIQUES ET DE GÉNIE INDUSTRIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION
DU DIPLÔME DE PHILOSOPHIÆ DOCTOR (Ph.D.)
(MATHÉMATIQUES DE L'INGÉNIEUR)
AVRIL 2001

©Aboubacar sidiki Sambou Bagayogo, 2001.



**National Library
of Canada**

**Acquisitions and
Bibliographic Services**

**395 Wellington Street
Ottawa ON K1A 0N4
Canada**

**Bibliothèque nationale
du Canada**

**Acquisitions et
services bibliographiques**

**395, rue Wellington
Ottawa ON K1A 0N4
Canada**

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-60930-8

Canada

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Cette thèse intitulée :

GÉNÉRATION DE MAILLAGE DE PEAU
POUR GÉOMÉTRIES COMPLEXES PAR
DÉCOMPOSITION SPATIALE
HIÉRARCHIQUE

présentée par : BAGAYOGO Aboubacar Sidiki Sambou

en vue de l'obtention du diplôme de : Philosophiæ Doctor

a été dûment acceptée par le jury d'examen constitué de :

- M. REGGIO Marcelo , Ph.D., président
- M. CAMARERO Ricardo , Ph.D., membre et directeur de recherche
- M. FORTIN André , Ph.D., membre et codirecteur de recherche
- M. GUIBAUT François , Ph.D., membre
- M. BEALE B. Steven , Ph.D., membre externe

À toutes les personnes qui me sont chères.

Remerciements

Au seuil de ce travail, je tiens tout d'abord à remercier et à exprimer toute ma gratitude au professeur Ricardo Camarero, mon directeur de recherche, pour sa compréhension, son encadrement et surtout pour toute la *grande* liberté qu'il m'a donnée tout au long du processus. Cette liberté m'a permis de développer encore plus mon autonomie intellectuelle. Je remercie vivement le professeur André Fortin pour avoir accepté la codirection de la thèse et un seul mot pour vous deux: merci.

J'exprime ma reconnaissance au professeur Neil F. Stewart du département d'informatique et de recherche opérationnelle de l'Université de Montréal. C'est à travers son cours gradué de modélisation des solides que j'ai entendu pour la première fois les termes Quadtree, Octree, décomposition spatiale etc. et qui par la suite a accepté d'être membre de mon jury d'examen prédoctoral écrit et oral. dernier examen d'une vie étudiante, une épreuve stressante.

Ma profonde reconnaissance à mon ex. étudiant Sylvain Fourmanoit, un de mes étudiants les plus doués, le tout doublé d'un sens du respect et de politesse. Mes amitiés sincères à mon collègue, "Sir" Antoine T. Massoud un homme fiable et sur qui l'on peut compter. Remerciements à tous mes collègues anciens et présents du bureau A-520.38 du département de mathématiques que sont: l'homme gentil et sociable qu'est Donatien Jean-Marie Kouakou N'dri, le "docteur" Jean-Luc Guilbault, Jean-Phillipe Marcotte, Alain Lioret, Steven Dufour, Josée Morin, Marie-Claude Heuzey, Florent Mobiot. Salutations à mes collègues Roberto Marano et Éric Turgeon du département de mécanique et à mes collègues Li Jun et Éric Paquette de l'Université de Montréal.

Je remercie mes *boss* coordinateurs de cours, Mmes Carol Burney-Vincent, Marie Luce Genot, Mrs. Jean Claude Warmoes, André Dupont, Clément Frappier, etc. pour la confiance qu'ils m'ont accordée lors des différentes charges de cours que j'ai eu l'occasion et avec plaisir d'assumer. Remerciements à tous mes collègues chargés de cours Guy Jomphe, Richard Labib, Hatem Ben Amor, Pontien M'baraga. Je remercie tout le personnel du département de mathématiques avec une mention spéciale pour Benoît Forest.

Remerciements et amitiés sincères à Panteha Pantt Shadpour, Sophie Lerouge, Fatimm Hamade Khalil, Ira Elvira Stuetler, Béatrys Marie Vandebosch, Marie-Josée Rivest, Imby Mylène Moisan. Je termine en remerciant tous mes étudiants présents et passés de l'École Polytechnique de Montréal, je demeure convaincu que parmi eux sortiront certains des plus grands ingénieurs canadiens du siècle.

Résumé

Le succès des méthodes de résolution numériques, les progrès dans la Conception Assistée par Ordinateur (C.A.O), et le développement des outils de visualisation font que les méthodes de génération de maillage ont connu de nombreux progrès au cours des dernières années pour devenir aujourd'hui un sujet d'étude à part entière. Notre projet porte sur la génération de maillages de peau par décomposition spatiale hiérarchique, cette nouvelle tendance qui veut que le maillage *colle* à la physique du problème. C'est une exigence fondamentale lorsque l'on veut résoudre des équations à grand nombre de Reynolds. Le maillage de peau sera généré dans l'espace compris entre l'objet et la face des octants frontières en face de l'objet. Nous présentons dans cette thèse et pour la première fois une extension 3D des travaux de Smith et de Lechziner. Ce passage en 3D a exigé de notre part l'utilisation d'un langage de programmation de haut niveau (MapleV) pour accélérer la phase modélisation géométrique et de discrétisation des objets. Nous avons aussi développé des algorithmes de projections des faces des octants frontières en face de l'objet, d'algorithmes de calculs d'intersection et d'ordonnancement des faces de complexité algorithmique au plus polynomiale. L'informatique étant marquée depuis une dizaine d'années par l'introduction d'ordinateurs parallèles à capacité de calculs énormes, nous proposons l'utilisation de machines parallèles pour accélérer les calculs en profitant de l'indépendance géométrique des octants.

Abstract

Over the past ten years, mesh generation has been identified as one of the most challenging and key components in a variety of areas such as solid modelling, Computer Aided Design (CAD), graphical rendering, Computational Fluid Dynamics (CFD) and semiconductor device simulation. The trend in the current generation of meshes is for a given geometry to the mesh to adapt to the physical behaviour of the problem, in order to resolve the thin boundary layers, wakes, and other viscous regions which are characteristic of high Reynolds-number viscous flows. This thesis is dedicated to these critical aspects by extending in 3D the work previously done by Smith and Leschziner. The emphasis is on data structures, algorithms, geometrical intersection problems and projections. We present a grid generation which takes advantage of the spatial decomposition technique. A 3D geometry surface is modelling by using superquadrics and coded by MapleV programming language. An Octree grid is produced and a viscous layer-grid is inserted between the grid and the body surface by means of a polynomial projection technique. The use of state-of-the-art data structures, geometrical intersections and projection algorithms, reduce the costs associated with various search operation and the technique of spatial hierarchical decomposition is demonstrated on several geometries. For the future of this work, we propose the use of a supercomputer in which can use the geometrical independance of the octree in order to speed up the calculations.

Table des matières

DÉDICACE	iv
REMERCIEMENTS	v
RÉSUMÉ	vii
ABSTRACT	viii
TABLE DES MATIÈRES	ix
LISTE DES TABLEAUX	xiii
LISTE DES FIGURES	xiv
LISTE DES SIGLES ET ABRÉVIATIONS	xix
LISTE DES NOTATIONS	xx
LISTE DES ANNEXES	xxii
CHAPITRE 1 INTRODUCTION GÉNÉRALE, POSITION DU PROBLÈME ET MOTIVATIONS	1

CHAPITRE 2 OBJECTIFS DE NOTRE RECHERCHE	8
2.1 Objectif général	8
2.2 Objectifs spécifiques	8
 CHAPITRE 3 CONSIDÉRATIONS PRATICO-THÉORIQUES DU MAILLAGE	 11
3.1 Présentation industrielle du maillage	11
3.2 Présentation théorique et mathématique du maillage	12
3.3 Qualités d'un maillage	13
3.3.1 Qualités topologiques	13
3.3.2 Qualités physiques	14
3.3.3 Qualités géométriques	14
 CHAPITRE 4 ÉTUDE DES MÉTHODES DE GÉNÉRATION DE MAILLAGE	 16
4.1 Méthodes de Delaunay-Voronoi	16
4.1.1 Construction de la triangulation delaunay	17
4.1.2 Critique de la méthode	19
4.2 Méthodes de type frontales	20
4.2.1 Critique de la méthode	22
4.3 Méthode de décomposition spatiale	23
4.3.1 Idée de base de la méthode	23
4.3.2 Critique de la technique	25
4.4 Un condensé de quelques problèmes	25
 CHAPITRE 5 LES MAILLAGES HYBRIDES	 27
5.1 Position du problème	27

5.2	Historique des maillages hybrides	27
5.3	Génération de maillages hybrides	29
5.3.1	Méthodes de progression à partir de la surface	29
5.3.2	Approche de Smith et Leschziner	36
CHAPITRE 6 OUTILS TECHNIQUES POUR GÉNÉRER UN MAILLAGE HYBRIDE		39
6.1	Modélisation et calculs géométriques	39
6.1.1	Représentation des modèles géométriques	40
6.1.2	Opérations pour une modélisation par superellipses	40
6.1.3	Modélisation du fuselage	42
6.1.4	Structures associées au fuselage central	45
6.1.5	Calculs d'intersection	49
6.1.6	Projections	52
6.2	Outils techniques de la génération de maillage par décomposition hiérarchique	57
6.2.1	Une structure de donnée fondamentale: l'arbre	57
6.2.2	Les Quadtree et les Octree	57
6.2.3	Synthèse de la génération de maillage par Octree/Quadtree . .	61
CHAPITRE 7 MISE EN OEUVRE INFORMATIQUE		65
7.1	Bilan de conception du programme Octree	65
7.1.1	Les grandes étapes du programme	66
7.1.2	Diagrammes de flots de données du programme	66
7.1.3	Arbre de construction des modules du programme	67
7.1.4	Digramme hiérarchique des fonctions génériques du programme	68

7.2	Design conceptuel, mise en oeuvre informatique et évaluation des algorithmes	69
7.2.1	Détection et calcul d'intersection	71
7.2.2	Indexation générale des faces	71
7.2.3	La projection	73
7.2.4	Génération des couches structurées de peau	78
7.3	Les structures de données du programme	84
7.3.1	Les classes Array et V3D	84
7.3.2	L'algèbre linéaire en dimension trois et la classe V3D	85
7.3.3	Règles générales de stockage	85
CHAPITRE 8 QUELQUES EXEMPLES D'APPLICATIONS . . .		88
8.0.4	Le programme Octree appliqué sur le nez d'un avion	89
CHAPITRE 9 PERSPECTIVES ET TRAVAUX FUTURS		107
9.1	Proposition de récupération de coins concaves	107
9.2	Proposition d'une avenue pour un critère physique automatique d'arrêt	108
9.3	Proposition pour une accélération de la décomposition Octree	110
CONCLUSION		113
RÉFÉRENCES		116
ANNEXES		122

Liste des tableaux

6.1	Paramètres de description de la tête du fuselage.	44
6.2	Paramètres de descriptions du cylindre central du fuselage.	47
6.3	Paramètres de descriptions du cône arrière du fuselage	47
6.4	Représentations des courbes et des surfaces.	49
7.1	Structures de données à seconde dimension variable.	86
7.2	Les structures de données selon leur contenu.	86
7.3	Les structures importantes du programme Octree.	87

Liste des figures

1.1	Maillages structurés et non-structurés.	3
1.2	Vision schématique de la répartition de la vitesse et des tensions tangentielles totales dans un écoulement à couche limite le long d'une plaque plane.	4
1.3	Visualisation de la répartition de la vitesse et du gradient de vitesse du/dz	5
2.1	Illustration bidimensionnelle d'un maillage hybride.	9
3.1	Place du maillage dans la conception générale d'un avion.	12
3.2	Forme des éléments.	15
4.1	Maillage et son dual. Les polygones de Voronoï constituent un maillage. Si le maillage dual est obtenu en reliant par une arête les centres des polygones ayant un côté commun est une triangulation, alors cette triangulation est dite de Delaunay.	18
4.2	Critère de la sphère vide.	19
4.3	Illustration bidimensionnelle de la création d'un nouveau front de montée.	21
4.4	Séquence de numérotation de la représentation Octree 3-niveaux d'un object.	24

5.1	Exemple de Marcum (1995) faisant passer le nombre de triangle de 11 544 au niveau de la couche limite (maillage non-structuré pur, figure à gauche) à 5 772 quadrilatères (maillage hybride, figure à droite). . .	28
5.2	Schéma d'avance hyperbolique.	30
5.3	Phénomène d'enchevêtrement de prismes et technique de recession de prismes.	31
5.4	Effet de la contrainte d'alignement quand on passe de la figure gauche à la figure droite.	32
5.5	Illustration graphique des étapes de l'approche de Smith-Leschziner. .	38
6.1	Exemple de fuselage complet.	41
6.2	Exemple de surface biparamétrée.	41
6.3	Un exemple de visualisation d'une opération de Spinning.	42
6.4	Profil du cône de la tête.	43
6.5	Illustration graphique de la fonction de transition donnée par l'équation (6.23).	46
6.6	Décomposition et modélisation du fuselage Via MapleV d'un avion. .	48
6.7	Illustration de la notion d'approximation par points de l'intersection de deux surfaces.	52
6.8	Visualisation de la projection de points sur une courbe.	53
6.9	Vecteurs tangents r^u et r^w évalués en un point donné.	56
6.10	Structures, terminologie et opérations de base sur un arbre binaire. . .	58
6.11	Trois possibilités de calcul de voisinage et d'ancestrale.	58
6.12	Structure Octree.	60
6.13	Illustration générale de l'imposition de la différence d'un seul niveau de subdivision, le rapport 2:1.	62
7.1	Diagramme de flots de données de niveau 0 du programme Octree. . .	67

7.2	Diagramme de flots de données de niveau 1 du programme Octree. . .	67
7.3	Arbre de construction du programme Octree.	68
7.4	Arborescence des fonctions génériques de Octree.	69
7.5	Suite de l'arborescence des fonctions génériques.	70
7.6	Une visualisation des octants frontières en face d'un objet.	74
7.7	Une visualisation des faces des octants frontières en face d'un objet: cas du nez de l'avion.	75
7.8	Une visualisation des faces des octants frontières en face d'un objet: cas d'une sphère.	75
7.9	Représentation octree à deux niveaux de décomposition du nez de l'avion.	79
7.10	Un zoom des projetés des noeuds des faces des octants frontières en face de l'objet.	79
7.11	Une sphère représentée de manière précise uniquement par la trace de la projection des noeuds des faces des octants frontières en faces de l'objet.	80
7.12	Une autre visualisation de la projection des noeuds des faces des oc- tants de la frontière en face de l'objet.	81
7.13	Topologie prismatique du maillage de 3 couches de peau.	82
8.1	Nez avion modélisé par superellipses.	89
8.2	Nez avion dans sa boîte englobante en une seule itération du pro- gramme Octree. On remarquera 4 octants frontières et 4 octants non-frontières.	90
8.3	Elimination des quatres octants frontières et projection des noeuds des faces des octants frontières en face de l'objet.	91
8.4	Rotation de la figure (8.3).	92

8.5	On isole les faces des octants frontières en face de l'objet pour mieux observer la précision de l'algorithme de projection.	92
8.6	Génération de la couche interne de peau.	93
8.7	Génération de la deuxième couche de peau.	94
8.8	Visualisation des deux couches de peau. Dans cette figure nous avons enlevé le maillage extérieur.	94
8.9	Visualisation de la projection des noeuds des octants frontières en face de l'objet pour une décomposition hiérarchique de niveau 2. . . .	95
8.10	Isolation des faces des octants frontières en face de l'objet de la figure (8.9).	95
8.11	Figure illustrant la génération de la première couche de peau.	96
8.12	Une visualisation de côté de la structure d'ensemble comprenant le maillage de peau et le maillage extérieur.	97
8.13	Visualisation de la couche intermédiaire dans le cas de la génération de 3 couches de peau.	98
8.14	Illustration complète des 3 couches de peau et du maillage extérieur de la figure (8.13) en représentation fil de fer.	99
8.15	Une visualisation des barres de projection après la construction d'une première couche de peau.	100
8.16	Décomposition octree générale de tout le nez de l'avion.	100
8.17	Zoom de quatre couches de peau générées autour du nez.	101
8.18	Visualisation de quelques étapes de la génération de maillage de peau autour du nez de l'avion.	102
8.19	Visualisation de couches de peau, suite.	103
8.20	Simulation Octree pour une sphère.	104
8.21	Simulation Octree d'un objet composé d'une sphère et d'un cône. . .	105
8.22	Simulation octree tore-cylindre.	106

9.1	Illustration de la récupération d'un coin concave en face du front de projection.	108
9.2	Illustration d'un processus CFD typique.	111
9.3	Illustration de la décomposition parallèle de chaque octant par un processeur indépendant.	112
A.1	Illustration graphique du code MapleV ci-dessus.	125
A.2	Gradation de la discrétisation pour différentes valeurs de la macro commande <i>grid</i> sur un plan.	126
A.3	Gradation de la discrétisation pour différentes valeurs de la macro commande <i>grid</i> une sphère.	127
B.1	Famille de quart de superellipse pour différentes valeurs de ϵ pour $r_x=r_y=1$	130
B.2	Superellipses avec différentes valeurs pour ϵ_1 et ϵ_2	135
B.3	Superhyperboles simples avec différentes valeurs pour ϵ_1 et ϵ_2	137
B.4	Superhyperboles doubles avec différentes valeurs pour ϵ_1 et ϵ_2	139
B.5	Supertores avec différentes valeurs pour ϵ_1 et ϵ_2	140
B.6	Illustration en 2D de la notion de courbure pour différentes valeurs de α et k . On montre l'existence de quelques points de discontinuité si la courbure est trop forte.	144

Liste des sigles et abréviations

CAO	Conception Assisté par Ordinateur
FAO	Fabrication Assisté par Ordinateur
CFD	Computational Fluid Dynamics
NURBS	Non Uniform Rational B-Splines
SDRC	Structural Design Research Corporation

Liste des notations

Caractères usuels

$B_{i,j}$	matrice des points du polygone de surface
d	distance de départ pour trouver un point qui soit le plus proche de Q
e	distance minimum à trouver quand on cherche un point sur la surface paramétrique qui soit le plus proche de Q
∞	infini
J	matrice jacobienne
k	ordre des entités géométriques
k	ordre de la courbe paramétrique B-spline
L	longueur
$N_{i,j}(u)$	fonctions de base B-splines normalisées (fonctions de forme)
n	nombre de points de contrôle des entités géométriques
P	point de projection du point Q sur la courbe $p(u)$
P_i	points de contrôle
$p(u)$	fonction paramétrique
$P^u(u)$	dérivée de $p(u)$ par rapport à u
$P^{uu}(u)$	deuxième dérivée de $p(u)$ par rapport à u
Q	un point de l'espace à 2 ou 3 dimensions
r	dimension de l'espace paramétrique des entités géométriques
$r(t)$	fonction paramétrique
τ_u	gradient évalué en (u_i, w_i) dans la direction u .
τ_w	gradient évalué en (u_i, w_i) dans la direction w
$r(u, v)$	surface paramétrique
$r(u_i, w_i)$	point sur la surface paramétrique $r(u, w)$

$r(u_q, w_q)$	point de la surface paramétrique $r(u, w)$ qui soit le plus proche du point Q
$R_{i,k}$	fonction de base B-Spline rationnelle
s	premier paramètre d'une surface NURBS
S	vecteurs tangents en direction u
T	vecteurs tangents en direction w
t	deuxième paramètre d'une surface NURBS
\mathbf{u}	vitesse (vectorielle)
\mathbf{v}	vitesse (vectorielle)
\mathbf{x}	variable d'espace
V_i	points de contrôle
w_i	fonctions de pondération ou de forme
(x, y, z)	point en coordonnées cartésiennes
∇_w	incrément paramétrique de déplacement

Caractères grecs

η	viscosité pour un fluide newtonien
η_0	viscosité à cisaillement nul
Γ	arête (entité topologique)
Σ	sommet (entité topologique)
Ω	face (entité topologique)
Π	volume (entité topologique)

Indices

i	indice du point de contrôle dans une direction paramétrique
j	indice du point de contrôle dans une autre direction paramétrique

Liste des annexes

ANNEXE A	Une présentation de MapleV	123
ANNEXE B	Mathématique des superquadriques	129

Chapitre 1

Introduction générale, position du problème et motivations

De nombreux problèmes physiques se modélisent sous forme d'équations aux dérivées partielles. Lorsque l'on ne peut avoir directement de solutions analytiques, des méthodes numériques permettent d'obtenir une solution approchée. Ces méthodes développées depuis les années 1950, prennent en compte des modèles physiques qui se rapprochent de plus en plus du monde réel. Les domaines d'applications sont par exemple la mécanique des fluides, la météorologie, la résistance des matériaux, l'électromagnétisme etc. et peuvent conduire à la conception numérique de nouveaux produits. Dans le domaine de la mécanique des fluides, la simulation numérique complète et réduit les tests expérimentaux, diminuant ainsi très sensiblement les coûts de conception des produits, d'autant plus que les performances des ordinateurs et superordinateurs ne cessent de s'accroître et les prix ne cessent de baisser de sorte que le coût d'un calcul-type décroît constamment en fonction du temps. Grâce à cette diminution des coûts et à la rapidité de traitement, la mécanique des fluides s'est modifiée profondément. Une étude numérique permet d'une part de réduire le nombre d'essais en soufflerie et d'autre part, d'évaluer rapidement et quasi exhaustivement les performances de l'objet: la mise au point des formes définitives est donc accélérée et à moindre coût. La méthodologie générale d'une simulation numérique comprend les étapes suivantes: Le problème initial est discrétisé et la solution est recherchée en certains points, appelés noeuds, du do-

maine de calcul, puis elle est interpolée sur le reste du domaine. Pour réaliser cette opération, le domaine étudié doit être préalablement découpé en éléments simples dont les sommets sont les noeuds de la discrétisation: ces éléments sont des polyèdres, petits volumes de l'espace ou surfaces dans le plan. C'est le maillage du domaine de calcul et on peut y distinguer deux familles à savoir: *Le maillage structuré*: il est composé d'un ensemble de quadrangles ou de triangles (respectivement d'hexaèdres et de tétraèdres) empilés en couches régulières. Sa caractéristique principale est d'avoir un nombre constant de voisin par sommet. Le voisinage d'un noeud est alors connu par simple lecture matricielle, ce qui allège les structures de données. Cependant, il n'est adapté qu'à des cas simples puisqu'il nécessite un redécoupage par bloc qui devient fastidieux dès que le domaine à étudier devient complexe. De plus comme le nombre de voisins par sommet est constant, il n'est pas possible de procéder à des raffinements locaux, sans perdre la propriété de conformité. On ne peut donc augmenter le nombre de sommets dans les zones jugées importantes sans généraliser cet ajout à l'ensemble du domaine. *Le maillage non structuré*: il est en général, constitué de simplexes (triangles en deux dimensions et tétraèdres en dimension trois) mais peut également contenir des quadrangles en dimension deux et des hexaèdres ou des prismes pour la dimension trois. Le maillage dans ce cas est dit mixte. L'utilisation de simplexes présente de nombreux avantages. Ces polyèdres élémentaires permettent de découper des domaines très complexes tels que la zone comprise entre le fuselage, la nacelle et l'empennage d'un avion. Outre cette capacité de discrétiser des formes quelconques, le maillage de type non structuré offre également une grande souplesse pour contrôler localement le nombre de points de discrétisation.

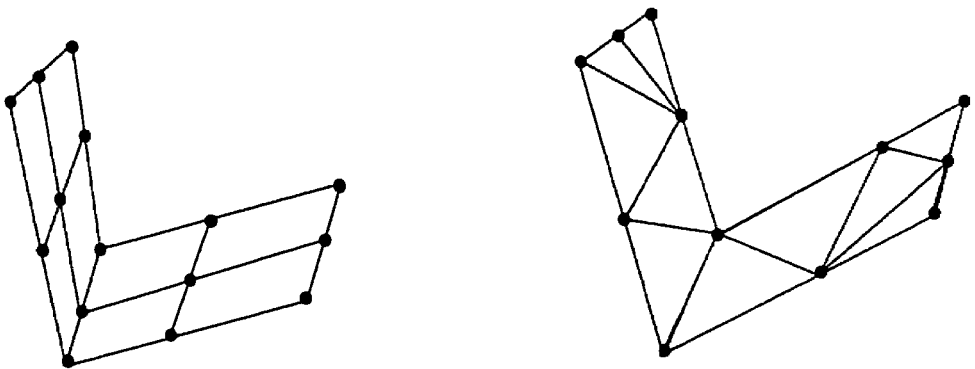


Figure 1.1: Maillages structurés et non-structurés.

Le succès des méthodes de résolution numériques, les progrès dans la Conception Assistée par Ordinateur (C.A.O.), et le développement des outils de visualisation, conduisent à simuler numériquement des problèmes physiques de plus en plus complexes. Afin d'obtenir des résultats plus précis, on ne se contente plus de découper le domaine de calculs en polyèdres élémentaires, mais il faut tenir compte de critères sur la discrétisation portant par exemple sur la forme des éléments, la densité des sommets dans une zone etc. Ces exigences ont permis aux méthodes de maillage de connaître de nombreux progrès au cours des dernières années pour devenir aujourd'hui un sujet d'étude à part entière. On est bien loin des méthodes manuelles de génération qui ne pouvaient s'appliquer que pour des géométries simples ! Cependant, aucune des techniques de génération automatique, telles qu'elles existent aujourd'hui, ne peut seule respecter l'ensemble des critères imposés à un maillage. Chacune d'elles, avec ses avantages et ses inconvénients, est plus ou moins adaptée selon la géométrie du domaine, les caractéristiques du phénomène étudié, le type ou le but de l'analyse effectuée. Notre projet porte sur la génération de maillages de peau par décomposition spatiale hiérarchique. C'est une nouvelle tendance dans la génération des maillages et qui veut que le maillage colle à la physique du problème pour mieux capter des phénomènes particulièrement complexes d'aérodynamisme externe liés à l'écoulement de fluide autour d'un obstacle. Les maillages hybrides constituent une voie mitoyenne permettant de tirer profit des avantages et des forces respectives des maillages structurés et non structurés, permettant ainsi de satisfaire plus rapidement les exigences de qualité pour mieux capter les caractéristiques fondamentales des écoulements à grand nombre de Reynolds. Ces écoulements qu'ils soient laminaires ou turbulents, présentent des couches limites minces près de la paroi. Dans un écoulement à couche limite on distingue deux domaines:

1. une zone proche de la paroi (surface du corps), de très faible épaisseur δ , appelée couche limite et où l'influence des forces de frottement (les tensions totales) est importante.
2. une zone éloignée de la paroi, appelée fluide libre, où l'influence des forces de frottement est négligeable.

Dans la couche limite, les tensions totales $\tau_{xx}(z)$, changent rapidement dans le plan normal à l'écoulement. La variation de la vitesse $u(z)$ est égale à zéro à

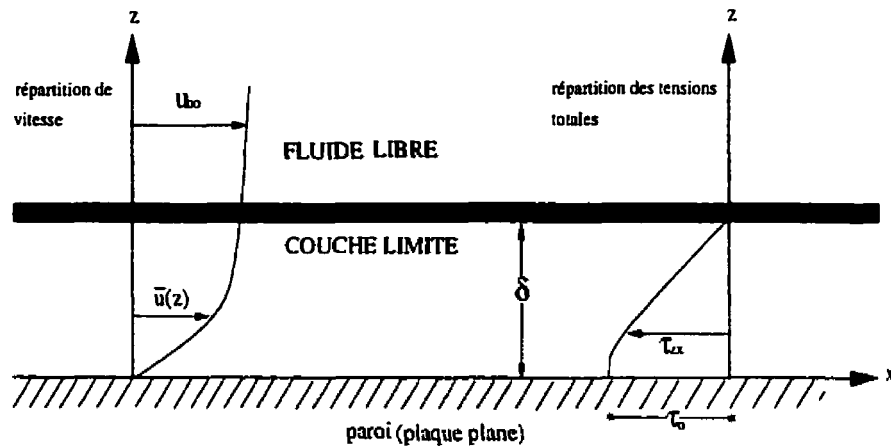


Figure 1.2: Vision schématique de la répartition de la vitesse et des tensions tangentielles totales dans un écoulement à couche limite le long d'une plaque plane.

la paroi $\bar{u} = 0$ car il y'a un phénomène physique d'adhérence des particules du fluides. Elle s'approche à une valeur confondue avec la vitesse de l'écoulement à fluide libre $\bar{u} = U_{\infty}$ à la distance δ . Par conséquent, dans la couche limite où a lieu la déperdition d'énergie, le gradient de vitesse du/dz (figure 1.3) est élevé et l'écoulement retardé. L'extension de la couche limite à partir de la paroi est définie

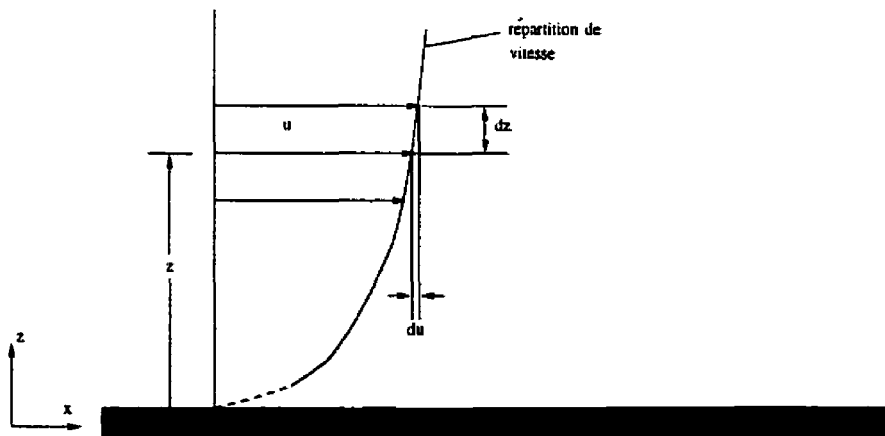


Figure 1.3: Visualisation de la répartition de la vitesse et du gradient de vitesse du/dz .

par l'épaisseur de la couche limite δ , mais il faut signaler que ces domaines ne sont pas nettement délimités, il y a transition entre eux et le flux de masse est continu. L'épaisseur de la couche limite δ croît le long de la paroi dans le sens de

l'écoulement du fluide. En partant de la description ci-dessus de la physique du problème, vouloir construire une discrétisation capable de capter ces phénomènes ce qui implique de placer un nombre suffisant de sommets pour décrire correctement les forts gradients présents dans ces régions. L'écoulement dans la couche limite est un phénomène très directionnel, une économie considérable en terme de nombres d'inconnues peut-être réalisée en utilisant des éléments très étirés, alignés avec les caractéristiques de l'écoulement, car la position des parois et des corps solides est connue pour un problème donné, il est possible dans une certaine mesure de construire à priori des maillages présentant des caractéristiques adéquates pour la résolution des équations. Le travail est organisé, articulé autour de 8 chapitres et de 2 annexes. Les chapitres du corps du texte introduisent au problème, alors que les annexes nous permettent d'approfondir et d'aggrandir notre largeur de vue ou compléments de réflexion sur ce qui se fait dans d'autres domaines de la science et du génie en rapport avec certains concepts et outils que nous présentons ici.

Les objectifs généraux et spécifiques de notre travail de recherche sont présentés dans le Chapitre 2. On y fait mention de la complexité des problèmes, des difficultés d'algorithmie, de structuration de données, de calculs d'intersection, de projection, d'ordonnancement des faces lors du passage en 3D. Nous insistons sur ce que nous voulons faire de manière générale et surtout de manière spécifique.

Le Chapitre 3 présente d'abord le maillage de manière pratique et de manière théorique. On y présente la place du maillage dans un cadre industriel et nous détaillons les étapes réelles de la discrétisation. Nous mettons en évidence les difficultés qui motivent nos travaux. Les définitions mathématiques de base et les qualités d'un maillage y sont présentées.

Dans le Chapitre 4 nous faisons un tour d'horizon des différentes méthodes de générations de maillage en présentant chaque fois la méthode et une critique des "bons" et des "mauvais" points de la méthode. Nous concluons ce chapitre par une synthèse des problèmes.

Chapitre 5: L'objectif de notre travail étant de produire un maillage hybride par la méthode des octree-quadtrees. Dans ce chapitre nous approfondissons

la problématique de la nouvelle exigence dans le domaine de la génération des maillages, à savoir, le besoin d'avoir un maillage qui *colle* à la physique du problème. Pour cela nous faisons d'abord un historique des maillages hybrides à travers une revue bibliographique ciblée depuis Kazuhiro Nakahashi en 1993 à nos jours. Nous présentons ensuite les deux grandes classes de méthodes de génération de maillages hybrides qui sont: la méthode de propagation en partant d'une triangulation de la surface et la méthode "top down", ou méthode de Smith et Leschziner dont le passage en 3D (une nouveauté) constitue notre travail de recherche. Cette approche est présentée aussi bien sur le plan algorithmique que sur le plan visuel. Pour faire un passage en 3D des travaux de Smith et Leschziner on a besoin d'outils techniques de modélisation géométrique des objets, les problèmes de calculs d'intersection, d'indexation, d'ordonnancement des faces et de projection etc.

Chapitre 6: Nous mettons en évidence les problématiques associées à la modélisation et à la représentation géométrique de nos objets dans un cadre CFD. Nous exploitons les résultats d'études de comparaison (présentés en annexe) des techniques de modélisation par superquadriques et par NURBS. Ensuite nous concluons ce chapitre par un choix éditorial de la technique retenue.

Chapitre 7: Dans ce chapitre nous faisons un bilan de conception du programme Octree suivant deux points de vues qui sont en fait complémentaires. Nous allons décrire le programme de manière dynamique à travers une vision DFD (Data Flow Diagram) et une vision statique à travers un diagramme hiérarchique des routines indispensables du programme. Dans ce chapitre nous passons aussi en revue le design conceptuel des algorithmes suivi de leurs mises en œuvre informatique et d'analyses de complexités. Ce chapitre se termine par un tableau de synthèse des structures de données du programme et des règles générales de stockage. Pour démontrer le fonctionnement du programme nous l'appliquerons quelques exemples de formes géométriques dans le corps du chapitre 8.

Les perspectives d'avenir et travaux futurs de ce travail sont présentés dans le chapitre 9. Les perspectives articulées autour de trois axes de recherche: un premier axe de recherche consiste à la récupération des coins concaves dans le cas de géométries de types industrielles. Dans un deuxième axe de recherche

nous proposons un critère d'arrêt automatique et physique liée à l'épaisseur de la couche limite, à travers une approximation de la couche limite via l'équation de Navier-Stokes. Un troisième axe de recherche que nous proposons vise à l'accélération des calculs via une parallélisation de la décomposition octree.

Chapitre 2

Objectifs de notre travail de recherche

2.1 Objectif général

L'objectif de notre travail de recherche est de développer une méthode originale pour la génération de maillages hybrides constitués d'un maillage de peau et d'un maillage espace. La peau c'est-à-dire la partie du maillage qui est directement en contact avec la surface de l'objet et avec une certaine épaisseur sera faite de couches structurées prises en "sandwich" entre l'objet et le reste du maillage comme illustré à la figure (2.1). Cet ensemble hybride doit satisfaire la nouvelle exigence des maillages, qui veut que celui-ci colle à la physique du problème. C'est une exigence fondamentale dans le cas de calcul numérique résolvant les problèmes de mécanique des fluides numérique de la couche limite. Ce sont des problèmes importants et fondamentaux qui traitent de l'écoulement au voisinage d'une paroi rigide pour de grands nombres de Reynolds.

2.2 Objectifs spécifiques

Nous visons comme objectif spécifique le passage en 3D, qui nécessite pour notre part des contributions au niveau de :

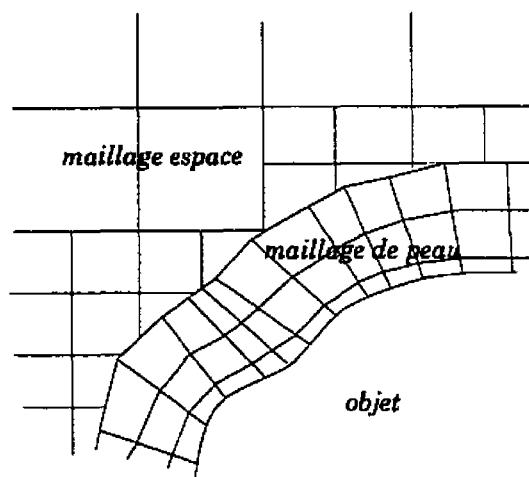


Figure 2.1: Illustration bidimensionnelle d'un maillage hybride.

1. La simplification mathématique des calculs d'intersections, un problème difficile qui fait l'objet par ailleurs de recherches actives et ouvertes. L'algorithme de calcul d'intersection que nous voulons développer devra obéir aux critères suivant:
 - (a) l'algorithme devra être précis au sens numérique du terme;
 - (b) robuste et capable de trouver toutes les portions de l'objet à être intersectées, objet qui peut avoir des parties disjointes;
 - (c) rapide pour utilisation dans un cadre interactif.
2. la projection, étape fondamentale pour générer des couches structurées de peau. L'algorithme de projection des noeuds des faces et des octants frontières en face de l'objet devra être entièrement automatique et performant.
3. la mise au point de techniques d'indexation générales explicitant les relations entre noeuds, arêtes, faces, volumes et positionnement des centroïdes.
4. des structures de données informatiques, que nous voulons simples, claires et générales. La méthode de la décomposition spatiale hiérarchique peut être une méthode informatiquement "explosive" et pour éviter cet aspect, le choix, le rôle, le profil, et l'organisation des structures de données informatiques devient un aspect fondamental. Les questions auxquelles nous devrions répondre sont:

- (a) Quelles types de structures de données devrions-nous choisir et quelles sont les opérations que nous pouvons appliquer sur ces structures (notion de fonctionnalité) et choix du support de stockage des données.
 - (b) Combien d'attributs de recherche spatiaux sont impliqués dans le problème? Il est bien connu d'une manière générale que certaines approches fonctionnent bien en 2D et sont catastrophiques en 3D.
 - (c) Quelle sera la complexité, l'accessibilité et la mise à jour des algorithmes? Dans le cadre de notre projet, nous voulons faire un choix d'algorithme au plus polynomial $O(1)$, $O(n \log(n))$, $O(n)$, $O(n^2)$) etc.
5. Au niveau de la modélisation géométrique des objets, nous voulons utiliser un langage de programmation de haut niveau, afin d'accélérer la phase de modélisation, de codage et de discrétisation des objets tridimensionnels.

Chapitre 3

Considérations pratiques et théoriques

3.1 Présentation industrielle du maillage

L'étude de quantités aérodynamiques locales ou globales lors de la conception d'un objet nécessite la résolution d'équations du type potentiel généralisé, Euler ou Navier-Stokes. Les résultats de ces calculs peuvent servir en particulier à redéfinir la géométrie des formes étudiées et à l'établissement d'une base de données de coefficients aérodynamiques de la forme étudiée. La génération d'un maillage tridimensionnel dans un milieu industriel se divise en général en trois étapes qui sont les suivantes :

1. Définition de la géométrie par le bureau d'étude des prototypes.
2. Création d'un maillage surfacique s'appuyant sur ces formes géométriques.
3. Génération de maillage spatial autour de l'objet, à partir du maillage de la surface.

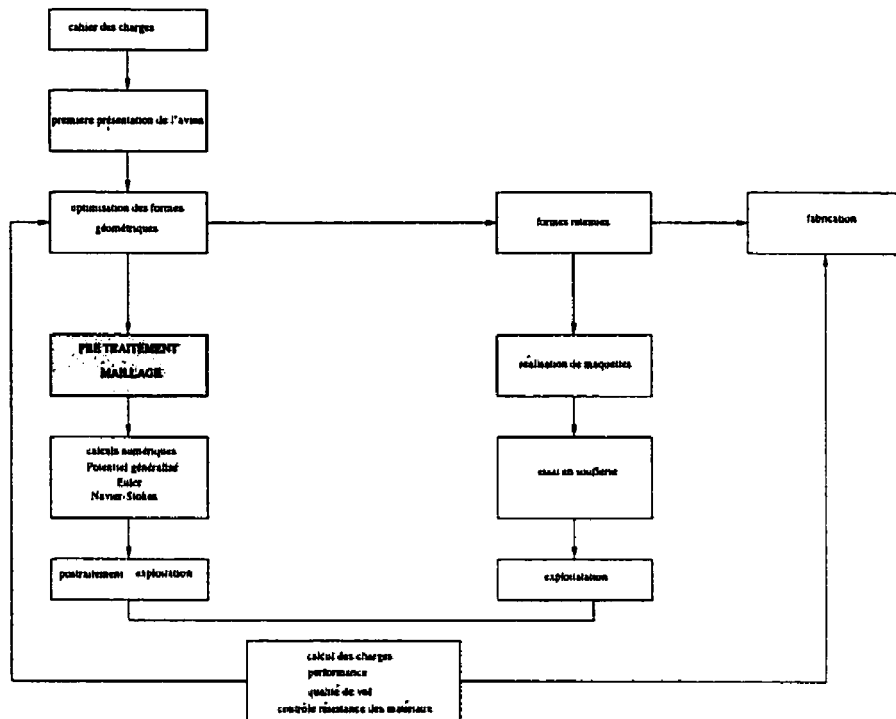


Figure 3.1: Place du maillage dans la conception générale d'un avion.

3.2 Présentation théorique et mathématique du maillage

De Coupez (1991), Castillo (1991), George et Borouchaki (1997) nous donnons la définition mathématique suivante d'un maillage: Soit un domaine Ω fermé borné qui se décompose en un nombre fini de polyèdres K de R^d . Nous considérons une décomposition T de Ω vérifiant:

1. $\bar{\Omega} = \bigcup_k K$ en chaque point du domaine Ω .
2. Tout élément K de T est un polyèdre de R^d d'intérieur non vide.
3. Les intérieurs de deux polyèdres de T distincts sont disjoints. Pour assurer la continuité aux interfaces entre les éléments géométriques K , il faut que :
4. Toute face d'un polyèdre K_1 de T soit face d'un autre polyèdre K_2 de T , soit une partie de la frontière Γ de Ω . Le domaine Ω est polyédrique, sa frontière

Γ est donc composée d'un ensemble fini de facettes qui correspondent à des faces de polyèdre, d'où la définition suivante: Un maillage d'un domaine Ω est une décomposition T de $\bar{\Omega}$ vérifiant les quatre propriétés ci-dessus.

3.3 Qualités d'un maillage

Les algorithmes de génération de maillage peuvent être complexes surtout si le domaine à mailler est compliqué. Mais que se soit au niveau de références que nous citons ici, ou ailleurs, on admet qu'outre les conditions de définition précédentes, le maillage doit également respecter des contraintes qui permettent d'améliorer les performances du code de calcul dont il est le support. Ces contraintes sont nombreuses et très souvent fonctions du code calcul utilisé. Le consensus général dans le milieu spécialisé définit trois types de critères pour distinguer les contraintes qui peuvent être imposées sur un maillage.

- Le premier type concerne la fidélité du maillage par rapport à la géométrie du problème.
- Le deuxième porte sur la densité des sommets en fonction des zones d'intérêts.
- Le troisième critère porte sur la forme des éléments.

Ces critères définissent respectivement les qualités topologiques, physiques et géométriques du maillage.

3.3.1 Qualités topologiques

Le maillage généré doit être le plus fidèle possible à la géométrie du domaine, qu'il soit convexe ou non. En particulier, les différents contours extérieurs et intérieurs qui apparaissent une fois que le maillage est généré doivent être aussi proches que possible des frontières originales.

3.3.2 Qualités physiques

La qualité de la solution obtenue par une méthode numérique du type éléments finis dépend de la finesse du maillage. Cela signifie que plus le maillage est fin (plus de concentration en éléments de ce maillage est élevée), plus précise sera la solution obtenue, ceci étant particulièrement vrai dans les zones où les gradients des variables du problème sont importants. Mais les coûts (temps de calcul, espace mémoire) augmentent généralement beaucoup plus vite que la finesse de maillage, et peuvent rapidement devenir prohibitif si l'on tente d'appliquer uniformément un maillage très fin. Une conséquence immédiate de ces considérations est que pour obtenir des solutions de bonne qualité sans trop augmenter la taille du problème considéré, il est important d'avoir une discrétisation beaucoup plus fine dans les zones où le phénomène physique varie plus fortement. La qualité physique d'un maillage est liée à la densité des sommets en fonction du problème physique à résoudre.

3.3.3 Qualités géométriques

Partons avec la méthode des éléments finis, une méthode d'approximation nodale et notons par u_h la solution approchée et par u la solution exacte, sur les éléments. Une erreur est donc commise: c'est l'erreur d'approximation. L'étude de la différence entre u_h et u sur chaque élément K indique la qualité de l'approximation qui est fonction de la régularité de K . La majoration calculée par Ciarlet et Raviart (1972) et reprise par la suite par différents auteurs est fonction du rapport $\frac{h_K}{\rho_K}$ où h_K est le diamètre de K (la plus grande des arêtes de K) et ρ_K la rondeur de K (rayon du cercle inscrit). Ce rapport est fonction de la forme des éléments: il est minimum dans le cas du triangle ou tétraèdre équilatéral. Afin d'améliorer la qualité de la solution numérique, il est recommandé d'avoir des éléments qui sont bien conditionnés, c'est-à-dire des éléments qui soient suffisamment réguliers. La qualité géométrique du maillage est alors quantifiée par un critère (coefficient de forme) qui mesure l'aplatissement des éléments. On peut décider dans certains cas d'étirer les triangles dans une direction donnée. Babuska et Aziz (1976), Bern et Eppstein (1992) montrent cependant qu'il ne faut pas que l'angle soit trop proche de 180 degrés.

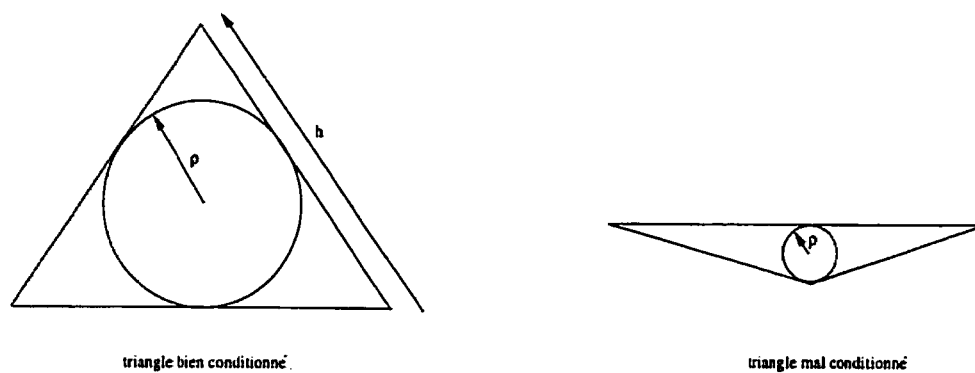


Figure 3.2: Forme des éléments.

Le but de tout algorithme de génération de maillage est qu'il soit en mesure de respecter les critères topologiques, physiques et géométriques. En plus il devra être entièrement automatique, efficace, facilement contrôlable et minimiser l'intervention de l'utilisateur.

Chapitre 4

Étude des méthodes de génération de maillage

Dans ce chapitre, nous présentons les principales techniques développées pour générer un maillage tétraédrique d'un domaine tridimensionnel. Nous distinguons trois groupes de méthodes en fonction des algorithmes et des structures qu'elles utilisent :

- Les méthodes de Delaunay-Voronoi qui utilisent la propriété du diagramme de Voronoi.
- Les méthodes de type frontales qui utilisent la frontière du domaine comme base de construction du maillage.
- Les méthodes de décompositions spatiales hiérarchiques qui subdivisent le domaine initial en sous domaines qui peuvent être maillés séparément.

4.1 Méthodes de Delaunay-Voronoi

Depuis le début des années 80, les propriétés des polyèdres de Voronoi sont utilisées pour générer des maillages et sont toujours le sujet de nombreuses publications. Leurs principes reposent sur les propriétés du diagramme de Voronoi qui

permet de construire un maillage en simplexes de l'enveloppe convexe d'un nuage de points. Cependant cette construction suppose que les points qui seront les sommets du maillage sont connus. Généralement, la solution obtenue contient trop peu de sommets pour qu'un calcul numérique puisse être effectué. Il faut alors enrichir la discrétisation du domaine de calcul. De plus, la méthode construit le maillage de l'enveloppe convexe d'un nuage de points ; la frontière n'est donc pas forcément respectée pour un domaine quelconque. Trois étapes sont requises pour générer le maillage d'un domaine quelconque :

1. Création de la triangulation Delaunay d'un nuage de point.
2. Création de points intérieurs pour obtenir un maillage de Delaunay utilisable par le calcul numérique.
3. Modification éventuelle du maillage pour respecter la frontière du domaine.

4.1.1 Construction de la triangulation delaunay

Soit un espace affine euclidien E de dimension d et $X = (X_1, X_2, \dots, X_m)$ un nuage de m sommets distincts de E . Le polyèdre de Voronoï V_i , associé au sommet X_i , est l'ensemble de tous les sommets de l'espace E plus proches de X_i que de tous les autres sommets de X

$$V_i = \{x \in E \text{ tel que } d(x, x_i) \leq d(x, x_j) \quad \forall j = 1, \dots, m\}. \quad (4.1)$$

Les polyèdres de Voronoï sont convexes, d'intérieur non vide et forme un recouvrement de E tel que deux polyèdres distincts ont une intersection soit vide, soit réduite à un point, à une arête, ou à une face. L'ensemble des m polyèdres constituent le diagramme de Voronoï. Son dual, obtenu en reliant les points X_i dont les polyèdres de Voronoï sont adjacents, est appelé triangulation de Delaunay. La triangulation T de Delaunay vérifie les propriétés suivantes :

1. Le cercle ou la sphère circonscrits à chaque polytope ne contient aucun sommet en son intérieur.

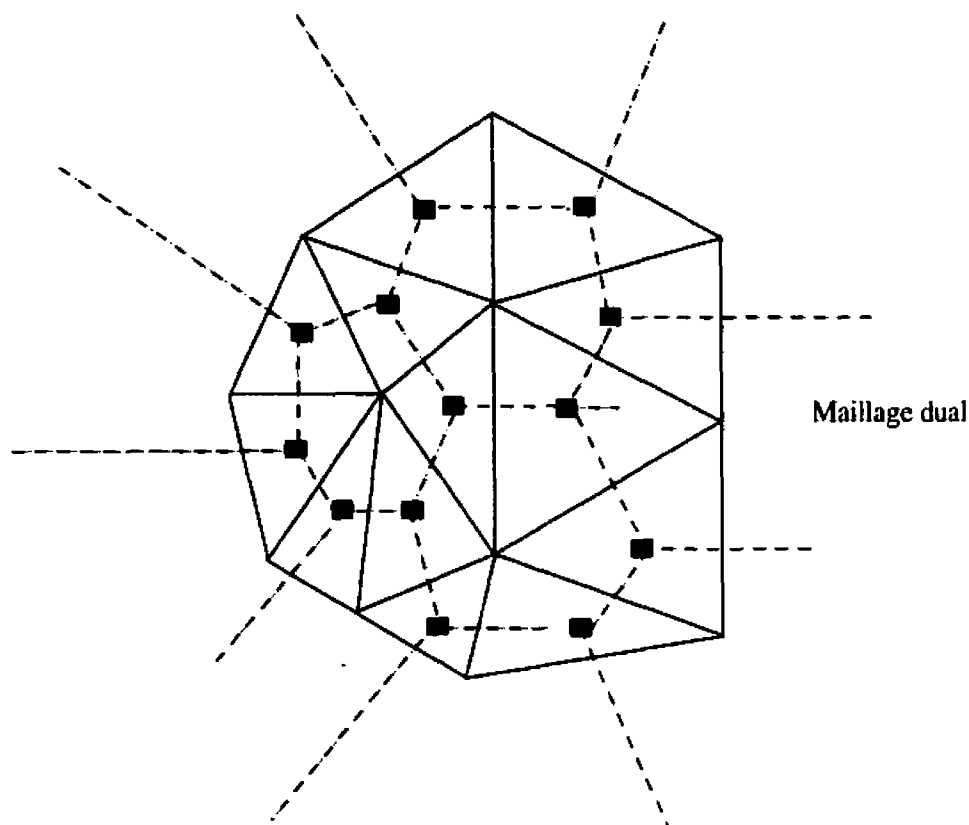


Figure 4.1: Maillage et son dual. Les polygones de Voronoï constituent un maillage. Si le maillage dual est obtenu en reliant par une arête les centres des polygones ayant un côté commun est une triangulation, alors cette triangulation est dite de Delaunay.

2. L'intersection de deux polytopes est soit vide, soit réduite à un point, une arête ou une face.
3. La réunion de T est égale à l'enveloppe convexe des points de X .
4. Si $d + 1$ sommets sont situés sur la même sphère, alors les polytopes sont des simplexes (triangles dans le plan, tétraèdre dans l'espace).

L'ensemble T apparaît donc comme un maillage de l'enveloppe convexe de points X . Lorsque la propriété 4 n'est pas vérifiée, les polytopes qui ne sont pas des simplexes doivent être décomposés en plusieurs simplexes. Dans ce cas il y a plusieurs décompositions. La propriété 1 de l'ensemble T de Delaunay est le critère de la

sphère vide, dit critère de Delaunay, qui en trois dimensions se formule de la manière suivante : la sphère circonscrite à chaque tétraèdre ne contient aucun autre sommet que ceux du tétraèdre. Un maillage respectant le critère de la sphère vide est appelé maillage de Delaunay . Différents auteurs ont montré que ce type de maillage en

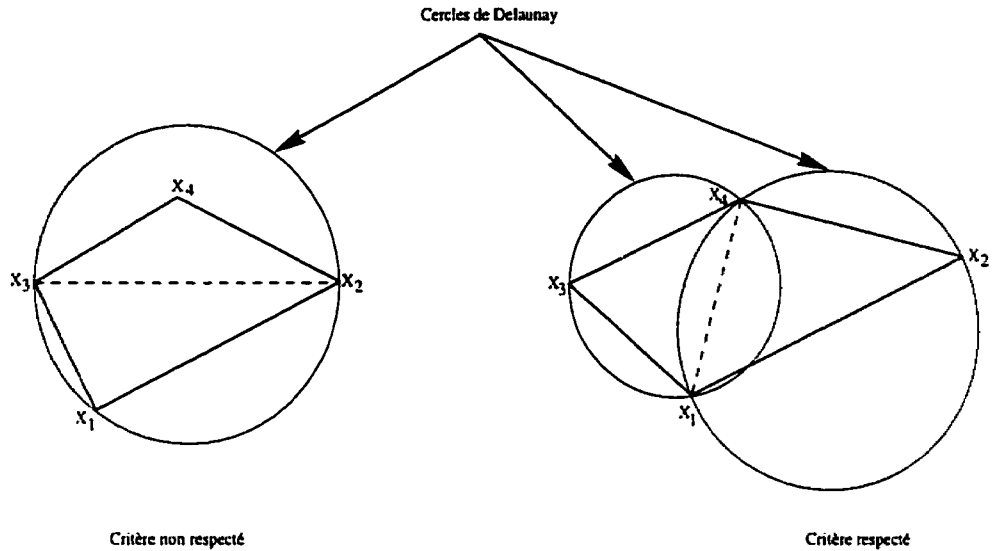


Figure 4.2: Critère de la sphère vide.

2D maximise sur une base locale l'angle minimum. Cette propriété est fondamentale pour garantir des triangles bien conditionnés, et a contribué au succès de la méthode. L'une des avancées théoriques dans le domaine est dû à Chew (1989) qui, dans un contexte de géométrie numérique, et à partir d'un algorithme basé sur la triangulation Delaunay, montre que n'importe quelle région polygonale 2D convexe peut être triangularisée de telle sorte que tous les triangles ont un angle au moins de 30 degrés, sauf si le polygone de départ à un angle plus petit que 30 degrés, et le coefficient de forme de la triangulation va être toujours borné.

4.1.2 Critique de la méthode

La généralisation des techniques Delaunay aux maillages 3D est difficile d'approche. En trois dimensions, la triangulation Delaunay est bien définie mais il n'existe pas de théorèmes sur l'optimalité des coefficients de forme, et il est connu que cette technique est capable de générer en 3D des éléments de pauvre

qualité à l'origine de certaines imprécisions et singularités numériques connues sous le nom de *Slivers*. De plus, on ne sait pas exactement comment respecter de manière robuste et rigoureuse les frontières d'un domaine tridimensionnel avec cette méthode. Néanmoins des travaux actifs sont en cours, et les tendances actuelles consistent à faire un premier maillage 3D du domaine suivi par des techniques heuristiques d'optimisation pour améliorer ou éliminer les éléments mal conditionnés afin d'assurer une bonne convergence des calculs numériques (Baker (1985), Joe (1995)).

4.2 Méthodes de type frontales

Contrairement à la méthode précédente où la connexion et la création des sommets du maillage sont résolues de manière indépendante, les méthodes de type frontal définissent simultanément les sommets et la façon dont ils sont reliés les uns aux autres. Une autre caractéristique de ces techniques est d'utiliser le maillage de la frontière du domaine (le maillage surfacique est une donnée du problème et il est composé de sommets et de facettes triangulaires) comme base pour créer le maillage espace. Elles assurent donc par construction le respect de la frontière du domaine. Ces méthodes sont progressives et elles consistent à grignoter le domaine par création d'un tétraèdre à partir d'une face triangulaire de la frontière courante. Une nouvelle frontière est alors redéfinie et le processus se poursuit ainsi. L'algorithme général en 3D de cette méthode s'énonce comme suit:

Tant qu'il reste des faces sur la frontière

- choisir une face triangulaire.
- créer un tétraèdre ayant ce triangle pour base.
- redéfinir la nouvelle frontière du domaine en supprimant les faces du tétraèdre qui étaient sur la frontière et en ajoutant celles nouvellement créées.

Les principales difficultés de cette technique sont de :

- Choisir un triangle qui servira de base à la création d'un nouveau tétraèdre.

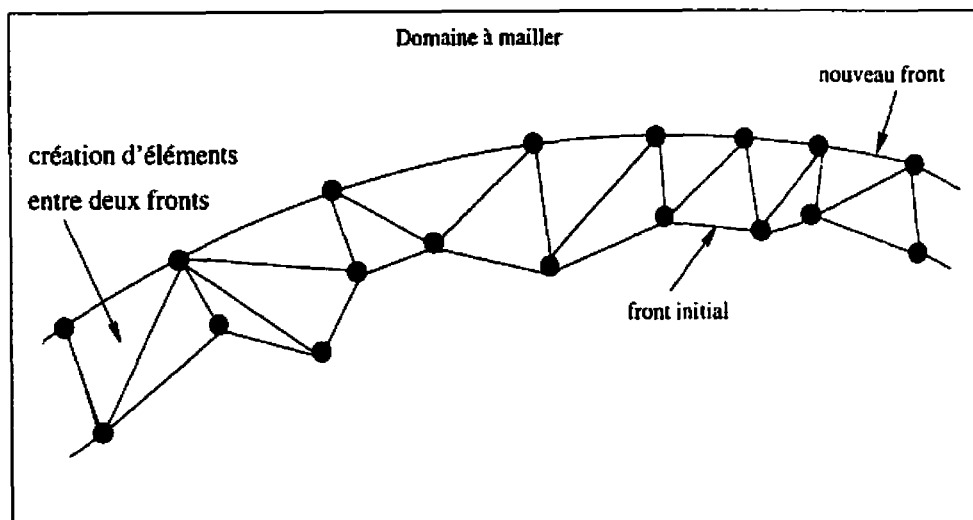


Figure 4.3: Illustration bidimensionnelle de la création d'un nouveau front de montée.

- Définir la position du nouveau point à l'intérieur du domaine lors de la création d'un nouveau tétraèdre.
- Vérifier que le nouvel élément créé est compatible avec le maillage existant, c'est-à-dire n'intercepte pas les autres éléments déjà créés.

Pour répondre au premier point, Golgolab (1989) propose d'établir un ordre de priorité sur l'ensemble des faces frontières susceptibles d'être sélectionnées. Il propose 5 solutions mais préconise les trois suivantes :

1. Parcours des triangles dans l'ordre décroissant de leur irrégularité.
2. Parcours des triangles dans l'ordre croissant de leurs dimensions.
3. Parcours des triangles dans l'ordre décroissant de leurs irrégularités pondérée par leur dimension.

Sa préférence est plutôt pour la troisième solution qu'il considère comme étant une moyenne des deux précédentes. Elle permet de débiter par les plus mauvais des plus petits triangles du front. Par ce choix, il espère créer un tétraèdre optimal en étant aussi peu contraint que possible par les tétraèdres voisins mais également

éviter la création de grands tétraèdres dans une zone contenant de petits triangles. Dans la méthode développée par Peraire et al. (1986), le triangle retenu est celui qui a la plus petite surface, ce qui correspond à la deuxième proposition de Golgolab. Ces choix sont bien sûr arbitraires et de nombreuses autres solutions peuvent être envisagées

4.2.1 Critique de la méthode

Contrairement à la technique précédente, les méthodes frontales assurent par construction même le respect de la frontière du domaine. De plus, elles s'adaptent simplement à la génération de maillage anisotropes étirés dans certaines directions, permettent ainsi la construction d'une couche proche de l'objet et constitué d'éléments écrasés dans la direction orthogonale à l'écoulement. C'est une propriété fondamentale dans le cas d'un calcul numérique résolvant les équations de type Navier-Stokes. Il suffit d'imposer un critère approprié aux triangles situés à une distance donnée de l'objet. Pour construire des maillages anisotropes Peraire et al. (1990) propose d'utiliser des paramètres contenus dans un maillage de référence appelé *background mesh*: les paramètres sont définis aux sommets du *background mesh* soit par l'utilisateur, soit par exploitation d'un premier calcul effectué sur le maillage. Un critère lié à chaque triangle frontière est calculé par interpolation des paramètres définis aux sommets de l'élément du *background mesh* qui contient le barycentre du triangle. La forme du tétraèdre est alors fonction du critère lié au triangle qui sert de base à sa construction.

Ces méthodes permettent à partir d'un bon maillage près de l'objet, qui ne présente que de faibles irrégularités de longueurs d'arêtes, de construire des éléments de bonne qualité principalement en début de traitement. Cependant des difficultés apparaissent dès que les conditions de départ ne sont plus idéales:

- Il est impossible de construire un bon élément à partir d'un mauvais triangle de base.
- Le tétraèdre idéal dans une région donnée de l'espace n'est pas toujours constructible à cause de tétraèdres voisins.

- Les disproportions entre les triangles frontières entraînent la rencontre de différentes parties du front de montée.

Pour éviter des problèmes informatiques et numériques il est alors nécessaire soit d'accepter la création d'éléments de qualité médiocre, soit de détruire des tétraèdres précédemment créés pour réussir à boucher les *trous* qui restent dans le domaine. Les boucles infinies de constructions/destructions deviennent alors possibles et doivent être étudiées.

4.3 Méthode de décomposition spatiale

La technique utilise des bases de données basées sur une arborescence quaternaire (2D) ou octale (3D), un adressage spatial de chaque élément et une structuration de l'analyse de façon hiérarchique. Ce sont des structures fondamentales qui sont utilisées dans d'autres domaines de sciences et de génie comme la compression d'image, la modélisation des solides, en intelligence artificielle, etc. Pour notre part, nous allons utiliser ces structures dans un contexte de génération de maillages hybrides, se situant dans la continuité des travaux qui existent dans le domaine et dont nous donnerons les références au fur et à mesure de la logique de l'enchaînement des idées.

4.3.1 Idée de base de la méthode

Pour mieux comprendre le principe à la base du fonctionnement de la méthode, il faut partir de l'idée de décomposition hiérarchique, utilisée surtout en informatique graphique. Cette science fait souvent appel à des structures hiérarchiques qui sont essentiellement de deux types tel qu'expliqué dans Samet (1990a,b). On a la hiérarchie dans l'espace objet et la hiérarchie dans l'espace image. La hiérarchie dans l'espace objet consiste en une décomposition des objets en leurs éléments constitutifs, méthode efficace pour résoudre des problèmes tels que le positionnement des objets dans l'espace, ou l'étude des mouvements d'objets les uns par rapport aux autres. La deuxième structure, c'est-à-dire la hiérarchie dans l'espace image s'obtient en décomposant non plus l'objet, mais l'image dans laquelle est enfermée

l'objet. On décompose le contenant et non le contenu. Cette décomposition peut s'effectuer de plusieurs façons. En 3D il s'agira de partir par exemple d'un cube et d'effectuer des décompositions successives, récursives en régions cubiques telles que montrées à la figure (4.4). Cette organisation est à la base de la méthode de la décomposition spatiale hiérarchique dont les quadrees (en 2D) et les octrees en (3D) sont des cas particuliers. Techniquement on utilise des bases de données basées sur une arborescence quaternaire ou octale et de là on applique un adressage spatial sur chaque élément, la conséquence étant alors une structuration de l'analyse de façon hiérarchique. La méthode Quadtree/Octree permet la mise au point d'algorithmes automatiques capables de traiter les géométries complexes. Cette technique offre des possibilités intéressantes dans le domaine de la génération de maillages hybrides 3D qui fait l'objet de notre étude.

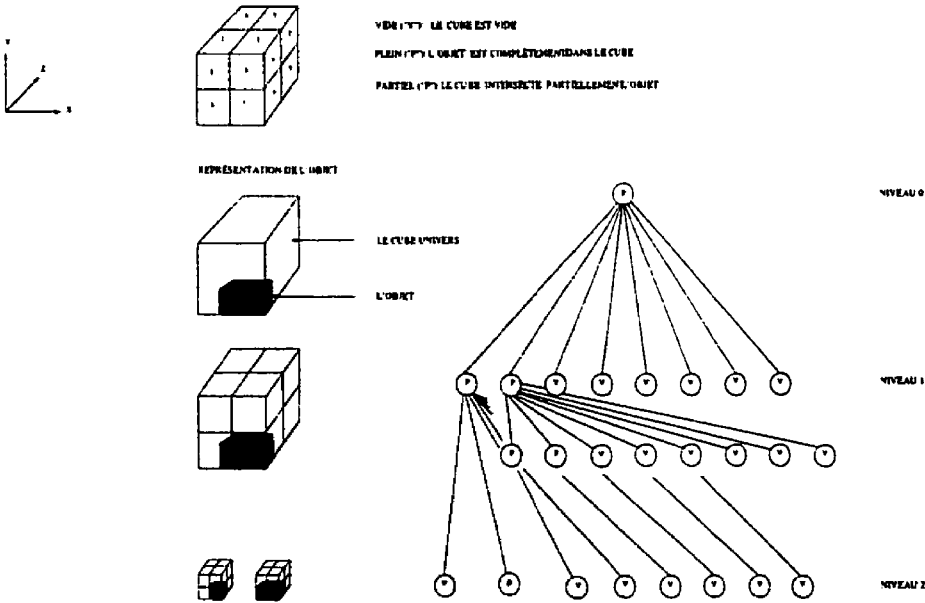


Figure 4.4: Séquence de numérotation de la représentation Octree 3-niveaux d'un objet.

4.3.2 Critique de la technique

La stratégie de décomposition spatiale hiérarchique est intéressante car elle permet :

- De découper le domaine en fonction des caractéristiques qui peuvent être géométriques, numériques (par exemple zones de forts gradients) ou encore physiques (objets multi-matériaux ou en plusieurs composants c'est-à-dire les objets complexes).
- De trianguler des domaines polyédriques 3D convexes ou non.
- Elle constitue une généralisation de la notion de triangularisation et est l'un des seuls choix possibles lorsque l'on travaille dans des espaces de dimension supérieure à trois (utilisation possible dans les techniques d'éléments finis espace-temps).

Les difficultés dans la méthode se situent aux niveaux suivants:

- Le calcul d'intersection pour le découpage et la classification des octants.
- Le calcul de la configuration des noeuds lors des tests d'appartenance, (les tests in/out).
- La reconstruction du domaine final dans le cas d'un objet complexe et des objets multimatériaux.
- Les maillages obtenus par cette méthode sont généralement de bonne qualité dans les régions éloignées des frontières. Au voisinage de celles-ci la situation est moins claire notamment en dimension trois.

4.4 Un condensé de quelques problèmes

Malgré l'existence de nombreux algorithmes, des avancées mathématiques et informatiques en terme de programmation et d'algorithmie, le problème de la génération d'un bon maillage est loin d'être résolu. En effet les différentes demandes

des utilisateurs engendrent de nombreuses difficultés qui nécessitent le savoir-faire et la patience du spécialiste du maillage. Afin de faciliter la tâche il est indispensable de mieux prendre en compte les besoins des utilisateurs comme:

- Le transport efficace de sommets d'une zone à une autre dans le maillage pour avoir une discrétisation suffisante aux endroits stratégiques.
- L'obtention d'un maillage correct après modification de la géométrie de l'objet considéré.
- Les problèmes de la représentation rigoureuse de la frontière de domaines complexes. La complexité des problèmes abordés, notamment en 3D rend l'utilisation et la maîtrise des algorithmes proposés dans la littérature fastidieuse et difficile.

Chapitre 5

Les maillages hybrides

5.1 Position du problème

Comme nous l'avons déjà signalé, aucune des techniques de génération automatique, telles qu'elles existent aujourd'hui, ne peut seule respecter l'ensemble des critères imposés à un maillage. Ainsi un bon maillage pour un programme qui résout les équations de type Euler peut être de mauvaise qualité pour un programme traitant les équations de type Navier- Stokes. Dans le premier cas (Euler), il est fondamental d'avoir des éléments qui ne soient pas aplatis alors que dans le second cas, on demande un élanement important des éléments dans la direction de l'écoulement. La technique des maillages hybrides trouve sa justification et sa motivation dans la tentative de résoudre cette contrainte. Dans ce chapitre nous faisons d'abord un historique des maillages hybrides, ensuite une étude synthétique des différentes approches qui existent pour générer un maillage hybride et nous terminerons par un choix éditorial de l'approche que nous avons choisie pour bâtir le maillage de peau en 3D.

5.2 Historique des maillages hybrides

Depuis la fin des années 80 les maillages hybrides ont commencé à attirer l'attention des chercheurs comme alternative à l'utilisation de maillages non-

structurés dans tout le domaine. Dans l'approche non-structurée pure et pour avoir une certaine précision numérique, il faut utiliser beaucoup de tétraèdres dans la région de la couche limite, augmentant ainsi la probabilité d'avoir des éléments de pauvre qualité connus sous le nom de *Slivers* à l'origine de certaines imprécisions et singularités numériques. On a alors un impact direct négatif sur la précision, le temps de calcul, et sur la robustesse des programmes informatiques dont le maillage est le support. Les premiers résultats concrets dans cette nouvelle approche de maillages hybrides datent de 1993 avec Nakahashi (1993), Kallinderis et Ward (1993). Marcum (1995), par l'exemple que nous illustrons ci-dessous montre que l'utilisation de quadrilatères à la place de triangles au niveau de la couche limite, a pour effet à précision numérique égale de réduire le nombre d'éléments, donc le temps de calcul. Kallinderis et al. (1995), montre l'avantage de l'utilisation de mail-

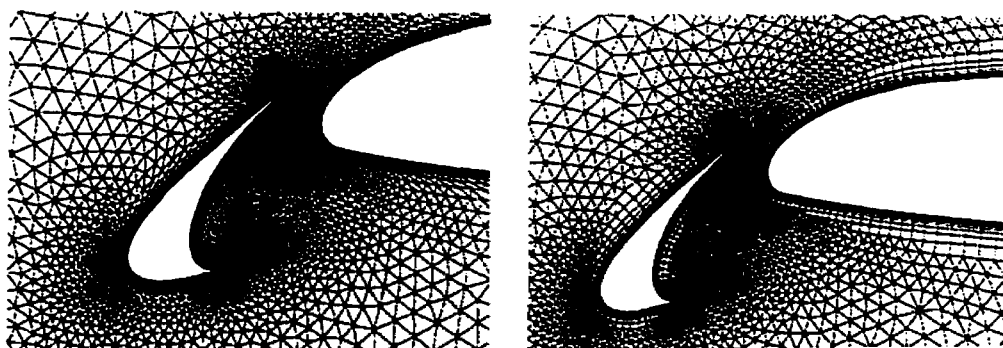


Figure 5.1: Exemple de Marcum (1995) faisant passer le nombre de triangle de 11 544 au niveau de la couche limite (maillage non-structuré pur, figure à gauche) à 5 772 quadrilatères (maillage hybride, figure à droite).

lages structurés par rapport aux non structurés pour mieux capter la physique du problème dans la couche limite, et réduire par la même occasion les temps de calculs. Pour des raisons d'espace mémoire, de vitesse d'exécution, et le désir d'avoir des éléments étirés à grand coefficient de forme, ils ont proposé l'utilisation d'éléments prismatiques structurés au niveau de la couche limite pour former le maillage de peau, zone où les effets visqueux sont dominants, et d'éléments tétraédriques à l'extérieur de la couche limite où l'écoulement est non visqueux. Des travaux existent dans le domaine de la génération de maillages hybrides et parmi ces travaux nous pouvons citer Coirier (1994), Parthasarathy et al. (1995), Khawaja et al. (1995), Chen et Kallinderis (1996), Pirzadeh (1996), Morris et Kallinderis (1996), Smith

et Leschzeiner (1996), Charlton et Powel (1997), Tchon et al. (1997), Minyard et Kallinderis (1998), Guibault (1998). Dans la section (5.3) nous faisons une synthèse des travaux de ces auteurs et des deux grandes approches qui existent dans le domaine.

5.3 Génération de maillages hybrides

On distingue principalement deux techniques: les méthodes de progression à partir de la surface et la méthode *top down* ou approche de Smith et Leschziner dont l'extension en 3D constitue l'objet de notre travail.

5.3.1 Méthodes de progression à partir de la surface

Nous allons distinguer principalement deux méthodes: La méthode de la progression hyperbolique et la méthode du zonage automatique de Guibault (1998).

5.3.1.1 Méthode de la progression hyperbolique

C'est une technique qui a été beaucoup étudiée au cours des dernières années par Kallinderis et Ward (1993), Khawaja et al. (1995), Kallinderis et al. (1995), Marcum (1995). La description ici suit globalement celle de Guibault (1998). La méthode d'avance hyperbolique consiste à placer les sommets du maillage en fonction du volume V des éléments désirés, et des angles θ des éléments. Pour un front initial Φ_0 à déplacer issu d'un maillage de surface, l'avance hyperbolique du front débute par la construction d'un vecteur de déplacement r_η à partir du vecteur tangent à la surface r_ξ tel que:

$$\frac{r_\xi \cdot r_\eta}{|r_\xi| \cdot |r_\eta|} = \cos\theta \quad (5.1)$$

et

$$|r_\xi| \times |r_\eta| = \Delta V. \quad (5.2)$$

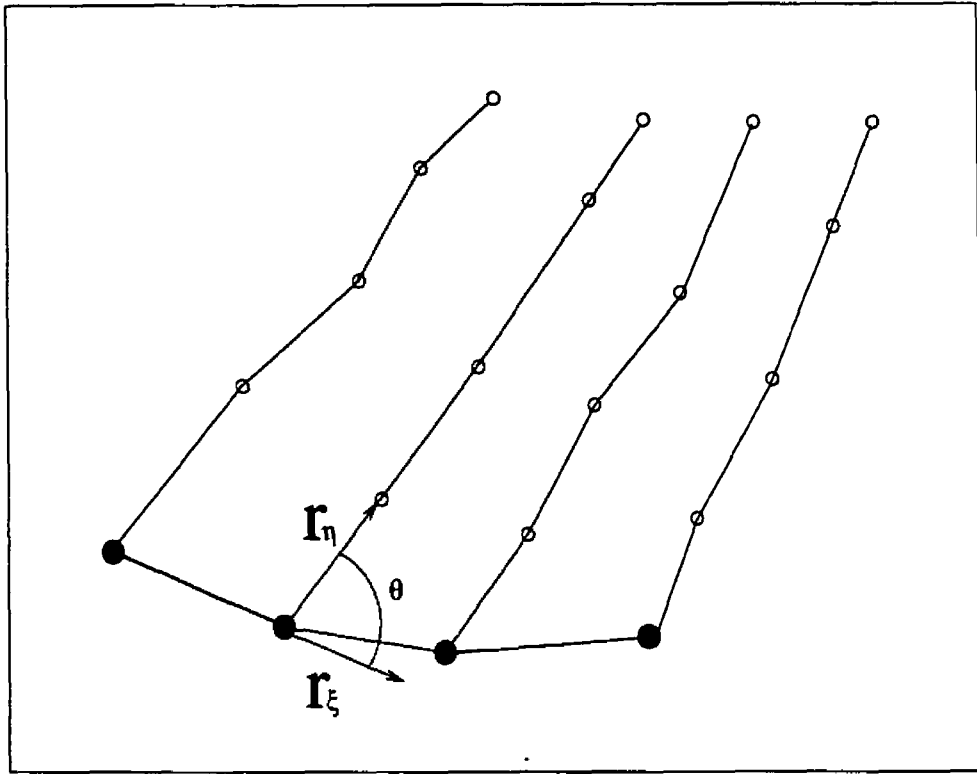


Figure 5.2: Schéma d'avance hyperbolique.

Ces conditions produisent un schéma d'avance hyperbolique du front initial:

$$r_{\eta+\delta\eta} = r_{\eta} + ds(\eta)Q(\theta)r_{\xi}$$

où $ds(\eta)$ est l'espacement désiré entre les couches d'éléments et $Q(\theta)$ est la matrice de rotation d'un angle θ . Chaque point est ainsi déplacé de r_{η} pour former un nouveau front Φ_1 , puis un second front Φ_2 etc. Cette propagation purement hyperbolique du front doit cependant être lissée pour éviter que des chocs ne se produisent durant la propagation. On peut utiliser l'opérateur elliptique suivant:

$$\alpha r_{\xi\xi} - 2\beta r_{\xi\eta} + \gamma r_{\eta\eta} = 0 \quad (5.3)$$

avec les conditions suivantes:

$$\alpha = r_{\eta} \cdot r_{\eta} \quad (5.4)$$

$$\beta = r_{\xi} \cdot r_{\eta} \quad (5.5)$$

$$\gamma = r_{\xi} \cdot r_{\xi} \quad (5.6)$$

Cet opérateur introduit à l'intérieur d'un algorithme prédicteur-correcteur et appliqué sur une couche (Φ_0, Φ_1, Φ_2) , permet de calculer une nouvelle position au front Φ_1 en quelques cycles de correction. La difficulté principale de cette approche réside dans la résolution des conflits se produisant lors de collisions entre des fronts issus d'objets distincts maillés simultanément. L'application de cet algorithme pour le calcul de la croissance de couches structurées autour de chaque élément fonctionne bien dans la mesure où les couches n'entrent pas en conflit les unes avec les autres et que l'on appelle phénomène d'enchevêtrement de prismes. C'est un phénomène que nous illustrons par la figure (5.3). Kallinderis

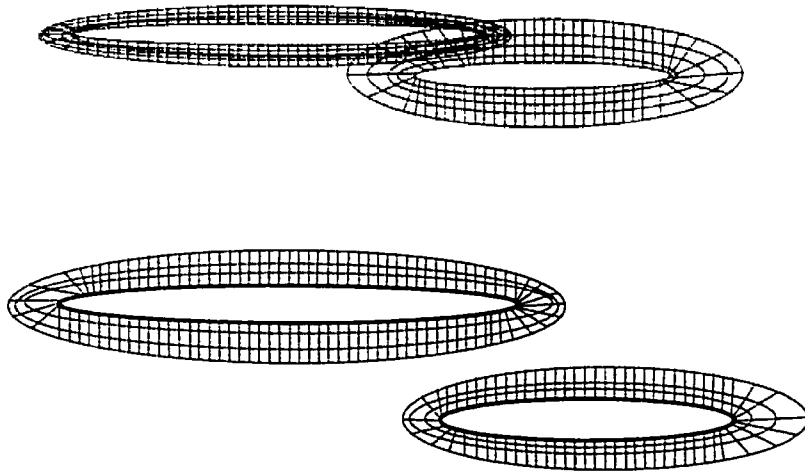


Figure 5.3: Phénomène d'enchevêtrement de prismes et technique de recession de prismes.

et al. (1995) présentent une approche systématique de détection des conflits et de repositionnement local des sommets sur les normales pour éliminer les collisions de fronts lorsqu'elles se produisent. Il reste que ce traitement des conflits consiste à diminuer l'épaisseur des couches structurées dans les régions posant problème. Un autre problème est le non-alignement de la ligne de marche lors de la progression

comme illustré par la figure (5.4). C'est un phénomène pouvant être à l'origine de la dégradation de la qualité des prismes générés. Pour résoudre le problème deux contraintes sont appliquées: l'une sur le positionnement des noeuds et l'autre sur l'angle maximal (30 degré) de tolérance de la déviation du vecteur de marche d'une couche prismatique à l'autre. La méthode de croissance hyperbolique se

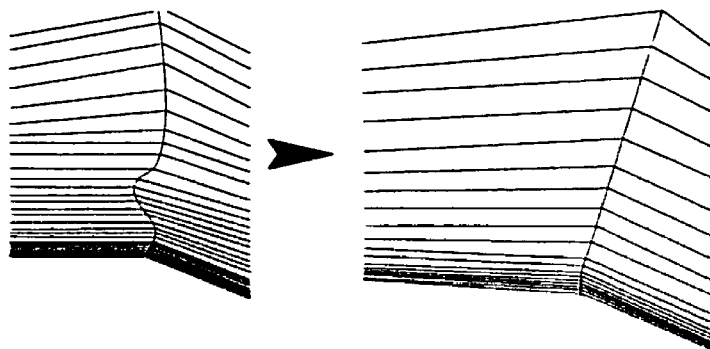


Figure 5.4: Effet de la contrainte d'alignement quand on passe de la figure gauche à la figure droite.

prête bien à la génération de maillages dans le cas de configurations d'écoulements externes, particulièrement lorsque la distance séparant les différents éléments géométriques sur lesquels doivent croître les peaux structurées est suffisante pour éviter les conflits entre les fronts. Par contre, cette approche ne propose pas de solution systématique au problème que représente l'interférence entre deux couches d'éléments structurés. L'approche par progression hyperbolique se prête bien aux cas de corps isolés, comme une aile constituée d'un profil simple, mais nécessite un traitement *ad hoc* pour des configurations plus complexes, comme celle d'une aile comportant plusieurs composantes.

Les problèmes liés à l'enchevêtrement des prismes peuvent être placés aussi dans le contexte des maillages *overset* (chimera). L'approche chimera dont les origines remontent à Joe Stegger dans les années 80 est très versatile et peut être attractive surtout dans le cas de corps en mouvement. La technique *overset* permet de mailler indépendamment la partition d'une configuration donnée et par la suite de les combiner pour former un maillage superposé. La méthode est efficace dans la mesure où chaque élément à analyser est bien séparé physiquement de ceux qui l'entourent, et que les maillages structurés restant, après la reconnection satisfont

aux critères minimums d'épaisseur de couche demandé par l'utilisateur. Les principales difficultés dans cette technique sont les suivantes:

- Il y a le problème de la précision de l'interpolation nécessaire pour transférer l'information d'un maillage à un autre.
- La méthode est difficilement envisageable dans le cas des écoulements internes, où toutes les parois solides sont nécessairement connectées les unes aux autres.
- La manière de combiner les différents objets qui ont été auparavant maillés séparément est laborieuse et nécessite l'intervention d'un spécialiste des maillages expérimenté.

La direction récente dans le domaine de la génération de maillages volumiques *overset* de Rogers et al. (2000) consiste à utiliser la méthode en conjonction avec un schéma de marche hyperbolique, le maillage résultant tend à être plus orthogonal avec un espacement de mailles plus lisse.

5.3.1.2 La technique du zonage automatique

La méthode de zonage automatique de Guibault (1998) consiste à partitionner le domaine de calcul en régions distinctes, maillable chacune séparément, tout en respectant les interfaces entre les zones de manière à assurer la continuité du maillage d'une zone à une autre. La partition se fait automatiquement, mais l'utilisateur peut contrôler le processus de croissance des zones, en indiquant au mailleur la liste des frontières qui doivent donner naissance aux nouvelles zones. Chaque zone peut être ensuite maillée selon un algorithme choisi par l'utilisateur, qui conserve ainsi le contrôle du processus de construction des zones structurées, mais également des aspects de génération de maillage dans chaque zone. Techniquement, le processus de maillage se décompose en deux phases distinctes: une phase de zonage automatique suivi par une phase de discrétisation de zone.

5.3.1.3 La phase de zonage automatique

Une face Ω_1 est bornée par une série d'arêtes Γ_j , et ses arêtes se rencontrent en une série de sommets Σ_i . Chaque entité topologique bornant la face peut-être

inscrite ou non dans la liste d'entités à déplacer. Déplacer une entité topologique consiste à créer une nouvelle entité topologique résultant de la partition de l'entité topologique originale, et à conserver dans l'entité originale la portion complément résultant de cette partition. Toutes les entités créées par ce processus ont la même dimension que l'entité topologique originale, c'est-à-dire que les entités topologiques de plus petite dimension se trouvant aux différentes interfaces ne sont pas considérées. Les grandes étapes du zonage peuvent-être résumées de la manière suivante:

1. le déplacement des arêtes en parcourant la liste des arêtes de la face, en vérifiant pour chacune si elle doit être déplacée ou non.
2. le déplacement des sommets, qui s'effectue par un bouclage sur tous les sommets Σ_i pour déterminer pour chaque sommet la nouvelle configuration topologique à y être associée. A cette étape toute l'information de configuration des entités topologiques a été passée des entités de dimension inférieure vers des entités de dimension supérieure, et le rôle de chaque entité dans la nouvelle topologie est déterminé. Toutes les entités déplacées ont été construites, il faut maintenant les connecter les unes aux autres ce qui constitue l'étape suivante.
3. la connexion des sommets. Cette tâche de reconnexion consiste à construire des entités topologiques de dimension supérieure issues des déplacements des sommets.
4. la connexion des arêtes qui consiste à rassembler l'information sur les arêtes construites lors des étapes précédentes afin de pouvoir construire la boucle externe de chaque nouvelle face.
5. la connexion de la face, et cette dernière étape consiste à construire la nouvelle face qui est le complément dans la face originale de toutes les faces construites lors du processus de déplacement. Cette face identifiée est construite en interrogeant toutes les arêtes des boucles de la face originale, et en insérant les arêtes elles-mêmes si elles n'ont pas été déplacées, ou leur arête déplacée si elles l'ont été.

À ces quatre étapes il faut rajouter en 3D deux autres étapes qui sont:

1. le déplacement des faces. Elle constitue sur le plan géométrique la partie la plus complexe de la mise en oeuvre du zonage. Il faut non seulement déplacer la face mais le faire en s'assurant que la nouvelle face insérée rejoigne les faces adjacentes à la face déplacée, de manière à former un nouveau volume qui soit fermé. La construction du support géométrique de la face déplacée se fait en cinq sous étapes:
 - (a) le déplacement des sommets le long des arêtes perpendiculaires à la face.
 - (b) le déplacement des arêtes dans les faces adjacentes à la face.
 - (c) le déplacement de la face comme tel.
 - (d) intégration des sommets déplacés aux arêtes déplacées.
 - (e) l'intégration des arêtes déplacées à la face déplacée.
2. la connexion des volumes est l'étape finale du zonage qui consiste à former une coquille pour les volumes complémentaires associés aux volumes affectés par le zonage.

5.3.1.4 La phase de discrétisation

La phase de discrétisation est appliquée à partir d'un algorithme choisi par l'utilisateur et s'effectue après la construction de la topologie et de la géométrie des nouvelles zones. Les opérations de discrétisation consistent à générer un maillage à l'intérieur de chaque zone, en fonction des paramètres fournis. La principale contrainte imposée au mailleur est qu'il puisse fournir des zones ayant des maillages conformes au travers de leurs interfaces, de façon à ce qu'un résolveur puisse être facilement utilisé sur le maillage résultant. Cette contrainte impose presque d'elle-même un algorithme de discrétisation où les frontières sont maillées avant l'intérieur des zones. Le processus de génération de maillage d'une zone associée à un volume se déroule de la façon suivante:

1. discrétiser chaque arête qui n'a pas été déjà été discrétisée.
2. discrétiser chaque face qui n'a pas déjà été discrétisée en respectant la discrétisation des arêtes.

3. discrétiser l'intérieur du volume en respectant la discrétisation des faces.

Le principal intérêt de cette approche de génération de maillage est qu'elle permet de produire des maillages hybrides sur des géométries complexes 3D comportant des couches minces d'éléments près des parois solides telles que requis pour la résolution des équations de Navier-Stokes. Elle peut être adaptée afin de construire des outils spécialisés de zonage en vue de produire des maillages multiblocs structurés.

5.3.2 Approche de Smith et Leschziner

Cette approche est basée sur la notion de hiérarchie dans l'espace image telle que déjà évoquée dans la sous-section (4.3.1) et qui consiste non pas à décomposer un objet, mais l'image dans laquelle est enfermée cet objet. La synthèse qui suit est basée sur les travaux de Smith et Johnston (1995), Smith et Leschziner (1996), Smith et Leschziner (1996). L'extension 3D, avec ce que cela implique d'algorithmes, de calculs d'intersection, de structuration des données et d'ordonnancement des faces constitue l'objet de notre travail. Nous synthétisons ces travaux d'abord d'un point de vue algorithmique et ensuite par une visualisation schématique des grandes étapes de la méthode par l'intermédiaire de la figure (5.5). Les grandes étapes de la méthode sont:

1. Définir les frontières du domaine et début de la décomposition quadtree (sous-figures (A) (E) et (F) de la figure (5.5)).
2. Générer les quadrants qui sont à l'intérieur du cadre englobant et qui intersectent l'objet.
3. Subdiviser les cellules non-homogènes et enlever les cellules qui sont à l'intérieur de la géométrie (sous figure C).
4. Itération jusqu'à satisfaction du niveau de subdivision quadtree désiré.
5. Se débarrasser des cellules frontières qui intersectent l'objet (sous figure (C)).
6. Application du critère 2:1 (sous-figure (B et C))

7. Lissage et projection suivant le principe de la plus courte distance des noeuds des quadtree frontières en face de l'objet (sous figure C et D).
8. Construction de la couche structurée de peau et de l'ensemble hybride en 2D (sous figure D).

L'algorithme suivant de calcul d'intersection en **2D** est proposé dans Smith et Johnston (1995), Smith et Leschzeiner (1996), Smith et Leschziner (1996):

1. Test quadtree_racine pour intersection avec l'objet.
2. Faire une descente dans l'arbre si test (1) est positif sinon ignorer cette branche.
3. Si le cadre englobant n'a pas d'enfants ou que le niveau de subdivision est atteint alors on arrête.
4. Si test (3) est positif subdiviser le quadrant.
5. Si test (3) négatif et que chaque branche locale a été visitée alors fin.
6. Si test (3) négatif et qu'il reste des branches non-visitées, retourner en (1).

Après avoir passé en revue les deux grandes approches qui existent dans la génération de maillage de peau, à savoir, les méthodes de progression à partir de la surface et la méthode de Smith et de Leschziner, nous pouvons constater que l'approche de Smith et de Leschziner basée sur une structure de donnée d'arbre est conceptuellement claire et est attractive à cause de sa simplicité et son grand potentiel pour l'automatisation. De plus, on peut faire varier le degré de la subdivision et comme conséquence de ce fait on peut avoir le niveau de précision désirée sur plusieurs ordres de grandeurs. Partant donc des remarques ci-dessous nous faisons le choix d'adopter cette approche comme méthode pour générer les maillages de peau en 3D, ce qui revient à faire une généralisation des travaux de Smith et Leschziner. Cette généralisation implique la résolution des problèmes d'acquisition rapide de la géométrie (qui peut prendre énormément de temps (Aftosmis (1995))), des problèmes de calcul d'intersection (un problème très délicat surtout en 3D), des problèmes de projection de points sur une surface (fondamental pour générer les couches structurées de peau) etc. Dans le chapitre suivant nous mettons en évidence

les problèmes centraux qui se posent lors du passage en 3D avant de faire quelques "choix" éditoriaux pertinents pour notre travail.

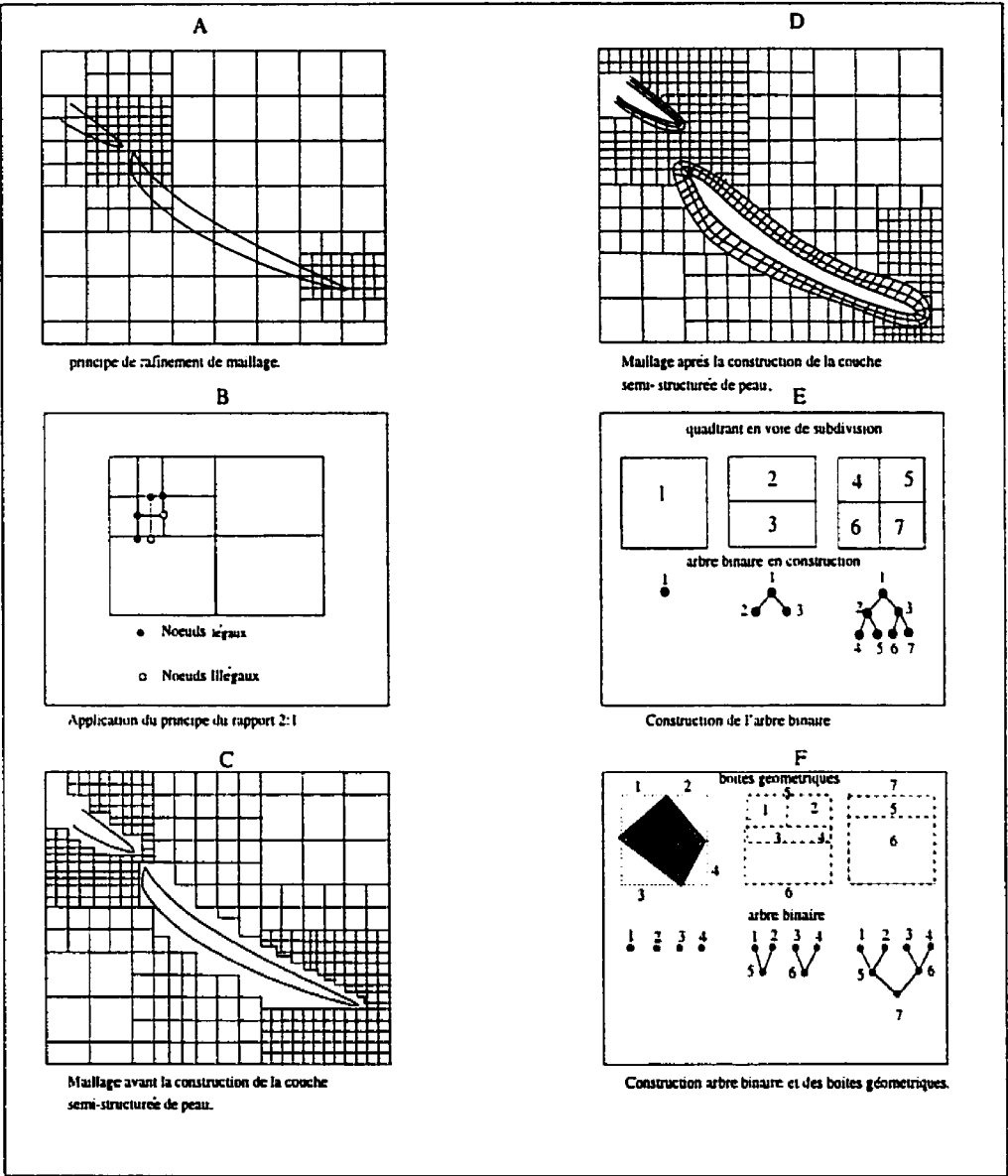


Figure 5.5: Illustration graphique des étapes de l'approche de Smith-Leschziner.

Chapitre 6

Outils techniques pour générer un maillage hybride

Dans le processus de génération de maillages hybrides (objet de notre étude) on a besoin d'outils techniques de modélisation géométrique, d'outils de calculs d'intersection, d'indexation, d'ordonnancement des faces et de projection, etc. En plus de ces outils techniques géométriques, nous avons aussi besoin d'algorithmes, de structures de données informatiques et de concepts techniques spécifiquement dédiés à la méthode de génération de maillage par Octree-Quadtree. Ce chapitre va être organisé autour des deux points que nous venons de signaler, et avant d'entrer dans le détail, nous remarquerons que la méthode de génération de maillages hybrides que nous proposons repose sur une **séparation stricte** entre le modèle géométrique et le mailleur. Cette séparation permet une grande souplesse et une versatilité dans la structure du mailleur lui conférant ainsi une grande capacité d'évolution.

6.1 Modélisation et calculs géométriques

La modélisation géométrique désigne la représentation des objets du point de vue de leurs formes et de leurs propriétés géométriques. Dans ce contexte la totalité des données requises pour représenter un objet ou un problème donné constitue le modèle objet, tandis que la partie purement géométrique de ces données définit

le modèle géométrique. Les techniques de stockage et de traitement des données géométriques sont relativement indépendantes des applications particulières. Des méthodes identiques peuvent donc être utilisées pour construire des modèles d'objets les plus divers dans les champs d'applications les plus variés.

6.1.1 Représentation des modèles géométriques

Les objets du monde industriel sont de formes très complexes sur lesquels des calculs de toutes sortes (aérodynamique, résistance des matériaux etc.) doivent être réalisés. La représentation informatique de ces objets conduit à rechercher des méthodes qui facilitent la manipulation des modèles. En tenant compte de l'annexe (B) nous avons fait le choix de représenter les objets via les superquadriques. Ce choix se justifie par le fait que les superquadriques permettent de modéliser une très vaste gamme d'objets qui sont utiles et dans beaucoup de domaines: infographie, vision robotique, en reconnaissance d'objets singuliers etc. Elles possèdent des propriétés mathématiques pratiques dont un nombre suffisamment petit de paramètres et une fonction ϕ , intérieur/extérieur dérivable sur tout l'intervalle des paramètres. La fonction ϕ , intérieur/extérieur est la clé de la puissance des superquadriques. Les superquadriques sont très versatiles et peuvent s'adapter à plusieurs domaines à l'aide de quelques additions ou modifications de leurs fonctions ϕ , intérieur/extérieur. Le langage MapleV présenté en annexe (A) a été utilisé pour la programmation et la discrétisation des objets.

6.1.2 Opérations pour une modélisation par superellipses

On se place dans un cadre aérodynamique et la démarche ici suit celle de Charlton (1997). Nous modélisons via les superellipses le fuselage d'un avion comme illustré par la figure (6.1). Le fuselage est composé de trois parties: la tête, le fuselage central avec éventuellement une extension protubérante comme on peut le constater dans les illustrations (c) et (d) de la figure (6.6). Pour la modélisation par superellipses de nos objets, on aura besoin de créer des surfaces, de faire des rotations et des extrusions. Les formes sont construites à l'aide de surfaces biparamétrées. Pour cela on établit une relation de fonctionnalité entre un point d'une surface (s, t) et

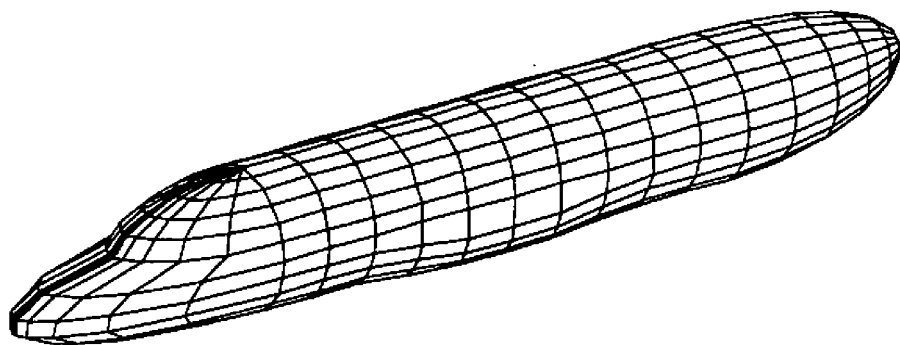


Figure 6.1: Exemple de fuselage complet.

un point (x, y, z) dans un espace 3D, tel que montré par la figure (6.2).

$$S = (x(s, t), y(s, t), z(s, t)). \quad (6.1)$$

Pour compléter la mise au point des formes on utilise les opérations d'extrusion et

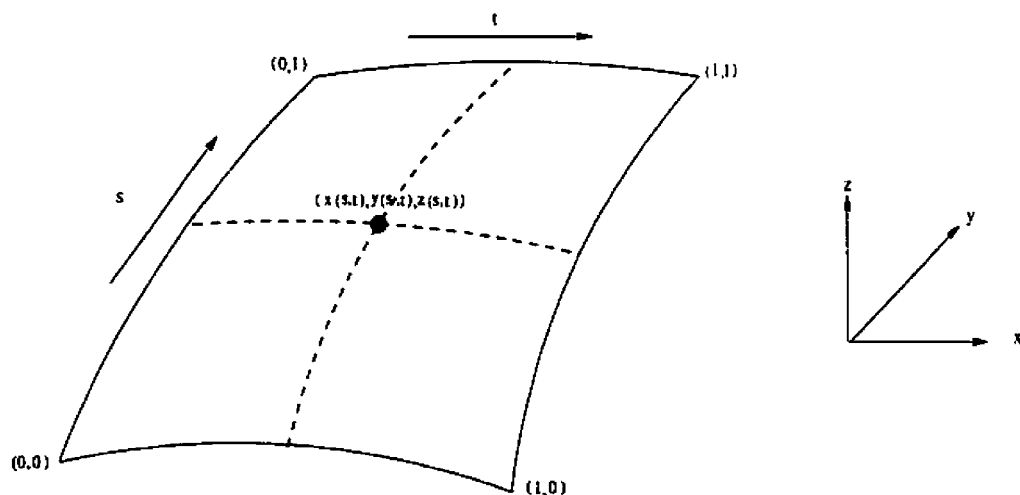


Figure 6.2: Exemple de surface biparamétrée.

de rotation (spinning) pour modéliser les autres composantes du fuselage comme le cône arrière et le cylindre central. Pour l'extrusion on part avec un profil donné par son équation paramétrique: $x = f(s)$ et $z = g(s)$ que l'on déplace de b unités dans

la direction de y pour $0 \leq s \leq 1$ et $0 \leq t \leq 1$ suivant l'équation (6.2):

$$x = f(s) \quad (6.2)$$

$$y = bt \quad (6.3)$$

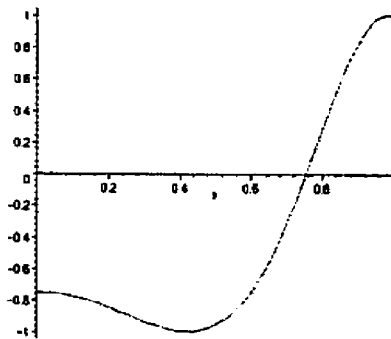
$$z = g(s). \quad (6.4)$$

Dans le cas du spinning, on démarre avec un profil $r(x)$ de longueur l par exemple et de paramètres $0 \leq s \leq 1$ et $0 \leq t \leq 1$ que l'on fait tourner autour de l'axe des x suivant l'équation (6.5), opération que l'on peut illustrer par la figure (6.3).

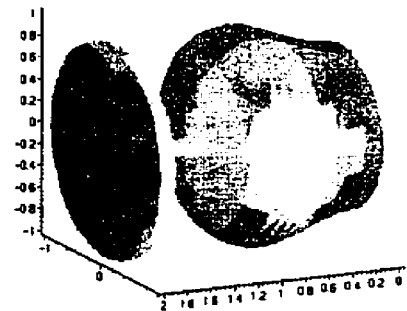
$$x = l(s) \quad (6.5)$$

$$y = r(x)\cos(2\pi t) \quad (6.6)$$

$$z = r(x)\sin(2\pi t). \quad (6.7)$$



(a) Profil avant Spinning.



(b) Résultat de l'application du Spinning.

Figure 6.3: Un exemple de visualisation d'une opération de Spinning.

6.1.3 Modélisation du fuselage

Le fuselage peut-être décomposé en trois surfaces séparées: la tête, le cylindre central et le cône arrière. La tête est une demi-ellipsoïde dont le grand axe varie dans

un cadre à double ellipse comme on peut le visualiser sur la figure (8.1). Le cylindre central est une surface biparamétrée que l'on peut localement modifier à l'aide des équations (6.22) et (6.23). Le cône arrière est tout simplement une ellipsoïde. Les autres paramètres techniques sont listés dans les tableaux (6.1), (6.2) et (6.3).

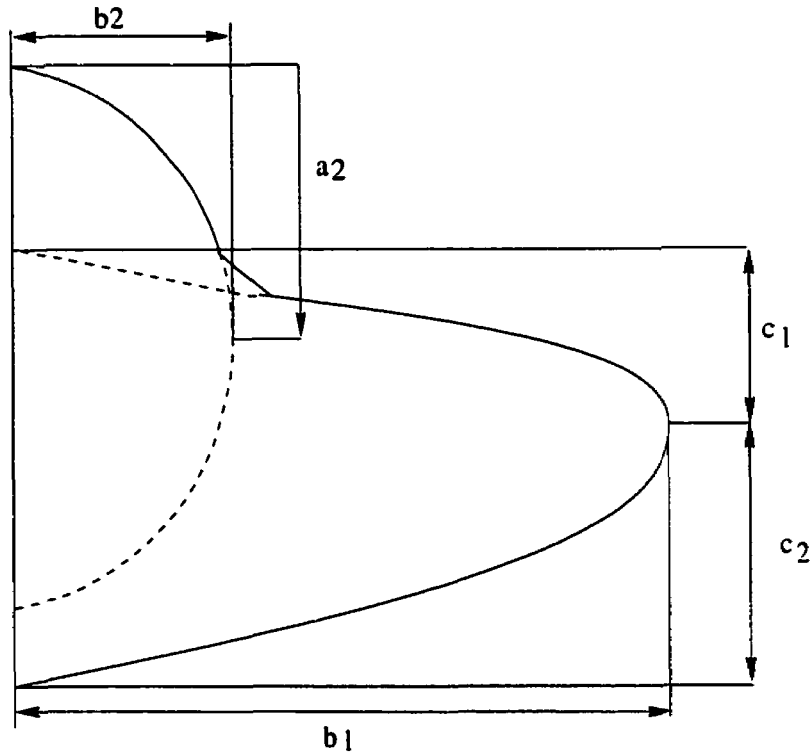


Figure 6.4: Profil du cône de la tête.

La tête est construite comme une ellipsoïde commençant par la section circulaire où elle intersecte le cylindre central. Le grand axe varie en fonction de l'axe z de manière à produire un profil aérodynamique qui, vu de côté, est en forme de double ellipse, comme illustré par la figure (8.1) et dont les paramètres sont listés dans le tableau (6.1). En jouant avec b_1 on peut allonger ou raccourcir la longueur de la tête et jouant avec le rapport b_2/a_2 on modifie la hauteur. La surface est construite à partir d'une série de courbes de niveaux pour des valeurs z constantes et pour la valeur d'angle $-\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}$. On définit une fonction par morceaux qui oscille entre deux valeurs basées sur le signe du troisième:

$$P(a, b, c) = \begin{cases} a & c < 0 \\ b & c \geq 0 \end{cases}. \quad (6.8)$$

Tableau 6.1: Paramètres de description de la tête du fuselage.

	Nose	
Parametre	Exemple	Description
w	6.26	largeur
a1	0.6	lower nose height
b1	12	nose length
c1	0.8	upper nose height
a2	1	cab
b2	9	cab
nPanelsR	32	discretization
nPanelsNose	32	Angular
x_nose y_nose	36 0	location

En se basant sur les paramètres techniques listés dans les tableaux (6.1), (6.2) et (6.3), et sur les formules mathématiques de la section (B.1) on définit l'ordre o par:

$$o = P(uorderD, lorderD, \theta). \quad (6.9)$$

On définit les *structures superellipses*, PowSin et PowCos par:

$$PowSin(\theta, o) = \operatorname{sgn}(\sin\theta)|\sin\theta|^o \quad (6.10)$$

$$PowCos(\theta, o) = \operatorname{sgn}(\cos\theta)|\cos\theta|^o \quad (6.11)$$

On définit le rayon z , par les expressions suivantes ci-dessous:

$$r = P(ruf, rl, \theta) \quad (6.12)$$

$$z^* = PowSin(\theta, o) \quad (6.13)$$

$$z = rz^*. \quad (6.14)$$

On construit la structure HalfEllipse de la manière suivante:

$$HalfEllipse(a, b, x_0, y_0, x) = \begin{cases} y_0 + b\sqrt{t} & t \geq 0 \\ -1 & t < 0 \end{cases} \quad (6.15)$$

où:

$$t = 1 - \left(\frac{x - x_0}{a} \right)^2 \quad (6.16)$$

et le rayon avant du nez, p' devient:

$$e_1 = \begin{cases} HalfEllipse(a1, b1, a1 - 1, 0, \frac{z}{r}) & z^* < a1 - 1 \\ HalfEllipse(c1, b1, a1 - 1, 0, \frac{z}{r}) & z^* \geq a1 - 1 \end{cases} \quad (6.17)$$

$$e_2 = HalfEllipse(a2, b2, 1 - a2, 0, \frac{z}{r}) \quad (6.18)$$

$$p' = \max(e1, e2). \quad (6.19)$$

Pour $-\frac{\pi}{2} \leq \alpha \leq \frac{\pi}{2}$, les coordonnées x et de y sont données par:

$$x = p' \sin \alpha \quad (6.20)$$

$$y = p' \cos \alpha. \quad (6.21)$$

6.1.4 Structures associées au fuselage central

La partie centrale du fuselage est modélisée comme un cylindre auquel on adjoit deux structures de transition. Les équations (6.22) et (6.23) permettent de changer le rayon du fuselage central à chaque point x , permettant ainsi la construction du carénage étendu. On peut se référer au tableau (6.2) pour la signification du symbolisme utilisé dans les stuctures de données ci-dessous.

$$s_1(f, a, t; s) = \frac{(f - a)(-tanh(\alpha x) + 1)}{2} + a. \quad (6.22)$$

Pour $\alpha = 4$, $f = 5$, $a = 4$ on a:

$$s_2(a, b, c, d; x) = \left\{ \frac{tanh(\alpha(x - b)) - tanh(\alpha(x - a)) + 2}{2} \right\} (d - c) + c. \quad (6.23)$$

pour $\alpha = 4$, $a = 3$, $b = 7$, $c = 0.7$ et $d = 1$. $0 \leq x \leq 10$, nous avons la figure (6.5).

$$r_u = s_1(ruf, rua, rug, \frac{x}{l}) \quad (6.24)$$

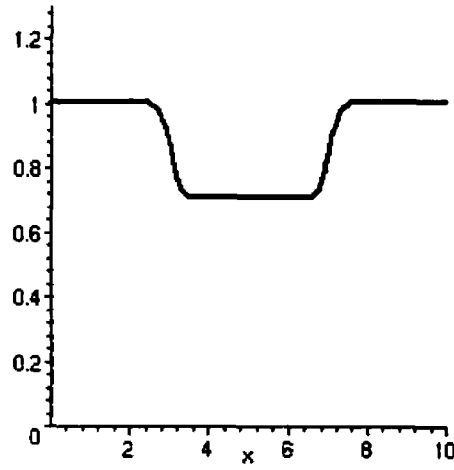


Figure 6.5: Illustration graphique de la fonction de transition donnée par l'équation (6.23).

$$r_l = s_2(rlA, rlB, rlC, rlD, \frac{x}{l}) \quad (6.25)$$

o_l et o_u

$$o_l = s_2(lorderA, lorderB, lorderC, lorderD, \frac{x}{l}) \quad (6.26)$$

$$o_u = s_2(uorderA, uorderB, uorderC, uorderD, \frac{x}{l}) \quad (6.27)$$

Pour chaque point x et pour chaque angle, $0 \leq \theta \leq 2\pi$, autour du cylindre central de rayon r on a:

$$r = P(r_u, r_l, \sin\theta) \quad (6.28)$$

et l'ordre o est:

$$o = P(o_u, o_l, \sin\theta) \quad (6.29)$$

En définitive nous avons:

$$y = r_y \text{PowCos}(\theta, o) \quad (6.30)$$

$$z = r \text{PowSin}(\theta, o). \quad (6.31)$$

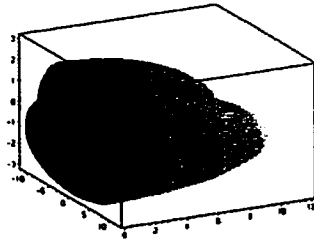
Finalement le cône arrière est une demi-ellipsoïde avec ses paramètres listés dans le tableau (6.3) et on peut se référer à la figure (6.6) pour une vision graphique et synthétique d'ensemble. Dans la section (A.2) du document on peut observer un exemple de programme en langage MapleV.

Tableau 6.2: Paramètres de descriptions du cylindre central du fuselage.

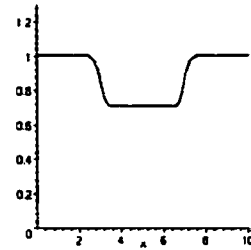
	Cylindre	
Paramètres	Exemple	Description
l	24.64	longueur du fuselage cylindre
ruf	3.08	forward
rua	3.08	aft upper radius
rug	3.08	upper
rla	0.4	lower
rlb	0.85	lower
rlc	3.08	lower
rld	3.08	lower
ralpha	16	lower length
nPanelsF	20	fuselage
uorderA	0.3	upper
uorderB	0.7	upper
uorderC	1.0	outer
uorderD	1	middle
uorderalpha	16	upper
lorderA	0.3	lower
lorderB	0.7	lower
lorderC	0.7	outer
lorderD	1.0	middle
lorderAlpha	16	lower

Tableau 6.3: Paramètres de descriptions du cône arrière du fuselage

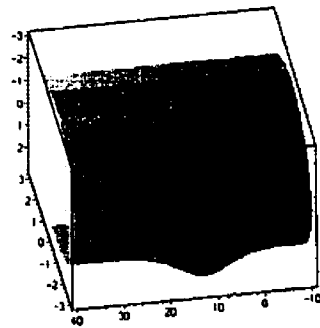
	Tailcone	
Parametre	Exemple	Description
ht	0.5	upper tail bias
lt	20.54	length of tail
nTailPanels	20	tail discretization parameter



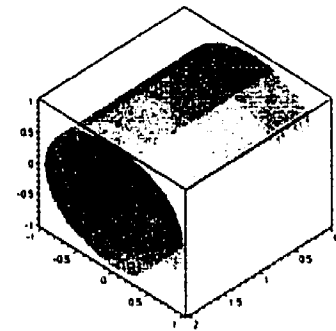
(a) Tête de l'avion modélisée par superellipses.



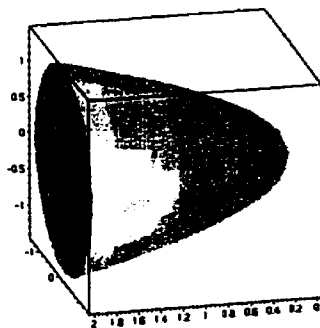
(b) Illustration graphique de la fonction de transition, (voir équation (6.23)).



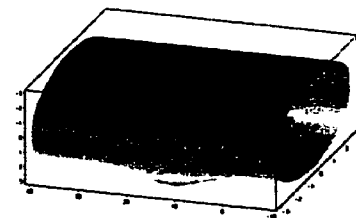
(c) Cylindre central après application de la fonction de transition.



(d) Cylindre central sans application de la fonction de transition.



(e) Demi-ellipsoïde représentant le cône arrière du fuselage.



(f) Illustration du cylindre central avec son carénage étendu.

Figure 6.6: Décomposition et modélisation du fuselage Via MapleV d'un avion.

6.1.5 Calculs d'intersection

Le développement d'algorithmes de calculs d'intersection précis, robustes et efficaces dans le contexte des courbes et des surfaces constitue un domaine de recherche actif et ouvert en CAO/FAO et un problème fondamental dans toute discipline qui utilise les méthodes de décompositions spatiales. Dans cette section nous analyserons le problème sous l'angle mathématique et analytique ensuite sous l'angle numérique et informatique.

6.1.5.1 Une classification mathématique des problèmes d'intersection

On admet avec Pratt et Geisow (1986) la classification suivante des problèmes d'intersection. De cette classification ces auteurs admettent qu'il y a trois types

Tableau 6.4: Représentations des courbes et des surfaces.

TYPE	COURBE	SURFACE
IMPLICITE 2D IMPLICITE 3D	$f(x,y) = 0$ $F(x,y,z) = 0$ $G(x,y,z) = 0$	$f(x,y,z) = 0$
PARAMÉTRIQUE	$\underline{r} = \underline{r}(t)$	$\underline{r} = \underline{r}(u,v)$
COURBE SUR UNE SURFACE PARAMÉTRIQUE $\underline{r} = \underline{r}(u,v)$	$h(u,v) = 0$ ou $\underline{r}(t) = \underline{r}(u(t)),v(t))$	

d'intersection possibles dans l'ordre de difficulté croissante suivante:

1. Intersection implicite/paramétrique. Si les équations de surfaces sont données par:

$$f(x, y, z) = 0 \quad (6.32)$$

$$\text{et } \underline{r} = \underline{r}(u, v) = [x(u, v), y(u, v), z(u, v)] \quad (6.33)$$

une simple substitution donne

$$f(x(u, v), y(u, v), z(u, v)) = F(u, v) = 0. \quad (6.34)$$

C'est le cas le plus simple, la courbe est représentée par une équation à deux variables, et elle est défini dans l'espace paramétrique de la deuxième surface.

2. Intersection Implicite/implicite. Les équations de surfaces sont données par:

$$f(x, y, z) = 0 \quad (6.35)$$

$$g(x, y, z) = 0. \quad (6.36)$$

où f et g sont généralement des fonctions non-linéaires. Dans ce cas la courbe d'intersection sera définie de manière classique par un système de 2 équations à 3 variables.

3. Intersection paramétrique/paramétrique. Les équations de surface sont données par:

$$\underline{r} = \underline{r}_1(u_1, v_1) \quad (6.37)$$

$$\underline{r} = \underline{r}_2(u_2, v_2). \quad (6.38)$$

Pour les points d'intersection des deux surfaces on a:

$$\underline{r}_1(u_1, v_1) - \underline{r}_2(u_2, v_2) = 0. \quad (6.39)$$

La courbe d'intersection sera représentée par un système de 3 équations à 4 variables u_1, v_1, u_2, v_2 , et les équations sont habituellement non-linéaires.

Analytiquement, on admet que la nature mathématique du problème d'intersection est relativement *simple* si une surface est représentée implicitement et l'autre paramétriquement. Ceci pose le problème de savoir s'il est possible de convertir, de passer d'un mode de représentation paramétrique à implicite ou vice-versa dans le cas où les deux surfaces sont du même type. Cette conversion conduit à un problème de type (1). Il est possible de réduire un problème de classe (2) à un problème de classe (1) par élimination algébrique d'une des variables, et de manière similaire de simplifier un problème de classe (3) par élimination de deux variables. Ces différentes possibilités sont réalisables à la conditions que toutes les fonctions concernées soient polynômiales ou des fonctions polynômiales rationnelles. La lourdeur des expressions fait que l'utilisation des langages de calculs symboliques comme Derives, Macsyma, ou MapleV (que nous présentons en annexe (A) peut être d'une grande utilité. Le but de l'utilisation de tels langages ou logiciels de calcul symbolique étant de réduire la partie technique du calcul algébrique.

6.1.5.2 Vision numérique et informatique

On calcule une approximation par points à la courbe d'intersection en fixant par exemple la valeur d'une variable comme illustré à la figure (6.7). Au besoin on impose des contraintes supplémentaires pour ensuite résoudre le système d'équations résultant. Guibault (1998) présente l'algorithme suivant de calculs d'intersection en 3D et qui consiste à discrétiser les entités géométriques avant de procéder aux calculs d'intersection. La méthode procède en 3 phases à savoir:

1. Identifier un point initial à une courbe d'intersection.
2. Si on trouve un point initial, suivre de triangle en triangle cette intersection sur les deux surfaces.
3. Convertir l'information accumulée lors de la poursuite des intersections en paramètres globaux sur les surfaces.

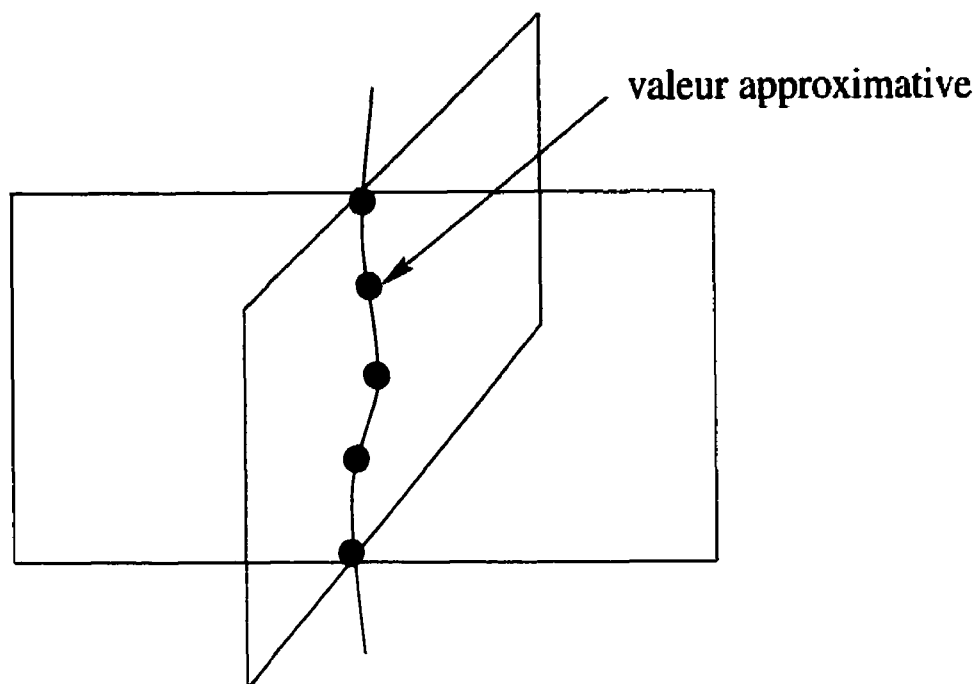


Figure 6.7: Illustration de la notion d'approximation par points de l'intersection de deux surfaces.

6.1.6 Projections

La projection P d'un point Q sur une courbe paramétrique $p(u)$, s'obtient par la construction du vecteur PQ en faisant en sorte que PQ devienne perpendiculaire au vecteur tangent P^u à la courbe $p(u)$ au point P . On peut écrire sous forme mathématique:

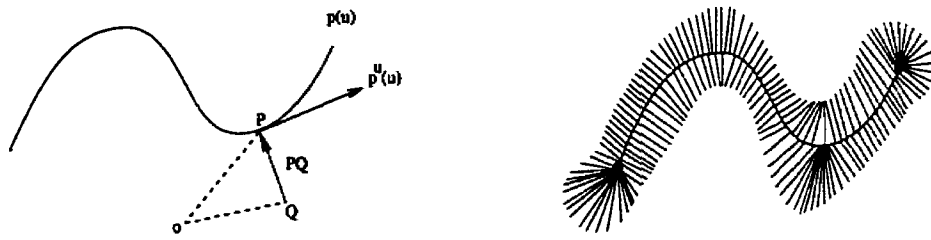
$$\overrightarrow{PQ} \cdot p^u(u) = 0. \quad (6.40)$$

Si, on place l'origine du système des coordonnées au point Q , l'équation (6.40) devient:

$$p \cdot p^u = 0. \quad (6.41)$$

En supposant $p(u)$ comme une courbe paramétrique cubique de la forme:

$$p(u) = a_3 u^3 + a_2 u^2 + a_1 u + a_0. \quad (6.42)$$



(a) Projection d'un point sur une courbe.

(b) Cas d'un ensemble de points.

Figure 6.8: Visualisation de la projection de points sur une courbe.

Cette transformation change $p(u)$ comme suit:

$$p(u) = a_3 u^3 + a_2 u^2 + a_1 u + a_0 - Q. \quad (6.43)$$

Mais les coefficients géométriques de la tangente $p^u(u)$ ne changent pas:

$$p^u(u) = 3a_3 u^2 + 2a_2 u + a_1. \quad (6.44)$$

Si on écrit l'équation (6.40) à partir de ses composantes, on a:

$$x(u)x^u(u) + y(u)y^u(u) + z(u)z^u(u) = 0. \quad (6.45)$$

En résolvant cette équation, on trouve les coordonnées du point P sur la courbe $p(u)$ qui est en général le point le plus proche du point Q . Mais, on doit cependant tester toutes les solutions pour ne pas prendre le point situé à une distance maximum. Pour résoudre cette équation, on peut trouver les racines à l'aide de la méthode de Newton-Raphson telle qu'expliquée dans (Piegl et Tiller (1997)). Pour cela on suppose une valeur u_i comme estimation initiale de la racine de la fonction $f(u)$ et après, on doit trouver une meilleure estimation u_{i+1} en procédant de façon itérative en utilisant la formule de mise à jour suivante:

$$u_{i+1} = u_i - \frac{f(u_i)}{f^u(u_i)} \quad (6.46)$$

où $f(u_i)$ et $f'(u_i)$ sont respectivement les valeurs de la fonction et de sa dérivée en $u = u_i$. Dans notre cas, $f(u)$ est donnée par l'équation (6.45) et sa dérivée $f'(u)$ est donnée par:

$$\begin{aligned} f''(u) = & x(u)x'''(u) + x''(u)x'(u) + \\ & y(u)y'''(u) + y''(u)y'(u) + \\ & z(u)z'''(u) + z''(u)z'(u). \end{aligned} \quad (6.47)$$

Les dérivées secondes dans l'équation (6.47) se calculent à partir de l'équation (6.43) d'après la formule:

$$p''' = 6a_3u + 2a_2. \quad (6.48)$$

On répète le processus avec l'équation (6.46) jusqu'à ce qu'on obtienne une réponse avec une précision désirée. La solution obtenue ci-dessus n'est pas toujours la bonne à cause des problèmes suivants:

- Il est possible qu'il existe plusieurs normales à partir du point Q sur la courbe dans l'intervalle $u \in [0, 1]$. Si c'est le cas, on doit trouver toutes les normales et sélectionner le point situé à une distance minimale du point Q .
- Il est possible qu'il n'existe aucune normale dans l'intervalle $u \in [0, 1]$. Dans ce cas on doit tester les extrémités de la courbe pour trouver le point situé à une distance minimale du point Q .
- Il est possible que le point Q soit situé sur la courbe, alors la norme du vecteur normale devient zéro et les points P et Q coïncident.
- La courbe peut être dégénérée (i.e. elle devient un point). Dans ce cas, il y a un seul point sur la courbe qui est le point recherché.

En 3D, pour trouver la projection d'un point $Q = (x_q, y_q, z_q)$ sur une surface paramétrique $r(u, w)$, on utilise l'inverse du Jacobien dans la procédure itérative de Newton-Raphson. Le but est de trouver le point $r(u_q, w_q)$ sur la surface paramétrique

$r(u, w)$, qui soit le plus proche du point Q . Comme précédemment on suppose une valeur (u_i, w_i) comme une estimation initiale de réponse. En introduisant les grandeurs suivantes:

$$d = Q - r(u_i, w_i) \quad (6.49)$$

$$e = r(u_q, w_q) - Q \quad (6.50)$$

où d et e représentent respectivement la distance de départ et la distance minimale à trouver. De même si on introduit la notation:

$$\vec{\nabla} u = (u_q - u_i) \quad (6.51)$$

$$\vec{\nabla} w = (w_q - w_i) \quad (6.52)$$

où $\vec{\nabla} u$ et $\vec{\nabla} w$ représentent les incréments paramétriques de déplacement dans l'espace. La fonctionnelle à minimiser est de la forme:

$$e = [r(u_q, w_q) - r(u_i, w_i)] - d \quad (6.53)$$

$$\approx (\vec{\nabla} u r^u - \vec{\nabla} w r^w) - d \quad (6.54)$$

où r^u et r^w sont les gradients (vecteurs tangents) évalués en (u_i, w_i) en direction u et w . On utilise la méthode des moindres carrés pour minimiser e et on obtient une expression de la forme:

$$\vec{\nabla} u = \frac{(r^u \cdot d)(r^w \cdot r^w) - (r^w \cdot d)(r^u \cdot r^w)}{(r^u \cdot r^u)(r^w \cdot r^w) - (r^u \cdot r^w)^2} \quad (6.55)$$

$$\vec{\nabla} w = \frac{(r^w \cdot d)(r^u \cdot r^u) - (r^u \cdot d)(r^w \cdot r^w)}{(r^u \cdot r^u)(r^w \cdot r^w) - (r^u \cdot r^w)^2} \quad (6.56)$$

À chaque itération, on doit calculer $\vec{\nabla} u$ et $\vec{\nabla} w$ et on trouve une meilleure estimation en mettant à jour u_i et w_i comme suit:

$$u_{i+1} = u_i + \vec{\nabla} u \quad (6.57)$$

$$w_{i+1} = w_i + \vec{\nabla} w \quad (6.58)$$

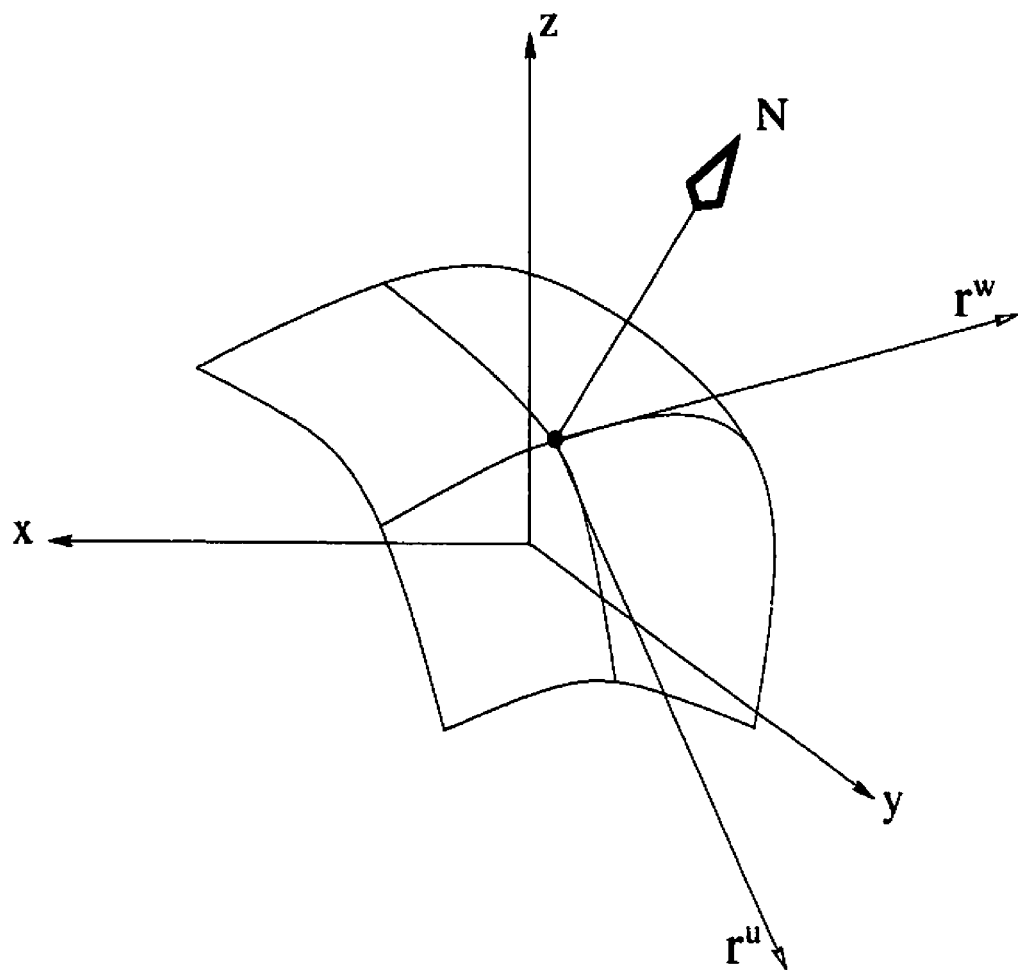


Figure 6.9: Vecteurs tangents r^u et r^w évalués en un point donné.

Si $|\vec{\nabla}u| \leq \epsilon$ et $|\vec{\nabla}w| \leq \epsilon$ pour une petite valeur donnée de la tolérance ϵ alors, on arrive à la solution:

$$u_q = u_i + \vec{\nabla}u \quad (6.59)$$

$$w_q = w_i + \vec{\nabla}w. \quad (6.60)$$

6.2 Outils techniques de la génération de maillage par décomposition hiérarchique

La technique utilise des bases de données basées sur une arborescence quaternaire (2D) ou octale (3D), un adressage spatial de chaque élément et une structuration de l'analyse de façon hiérarchique. La méthode étant basée sur une structure d'arbre, dans la sous-section (6.2.1) ci-dessous nous faisons une revue synthétique de cette structure de donnée fondamentale en terme de terminologie, de vocabulaire de base, de parcours etc.

6.2.1 Une structure de donnée fondamentale: l'arbre

Les arbres apparaissent régulièrement dans de nombreuses branches de l'informatique. On peut définir un arbre comme un ensemble d'éléments appelés noeuds dont un se singularise comme la *racine* liée par une relation dite de parenté induisant une structure hiérarchique parmi ses noeuds. Un noeud, comme tout élément d'une liste peut-être de n'importe quel type. Des figures (6.10) et (6.11) nous synthétisons et illustrons les concepts et terminologie de base relatif à cette structure de données fondamentale.

6.2.2 Les Quadtree et les Octree

Samet (1990a,b) a effectué à ce sujet une étude détaillée globaliste, descriptive, et une classification de ces structures de données spatiales en terme de binary trees, k-d trees, point optimized quad/octrees, bucket quad/octrees, regional quad/octrees etc. D'un point de vue historique, l'application des structures de données spatiales dans le domaine de la génération des maillages date des années 1980 avec Rheinbolt et Mesztenyi (1980), suivi par Yerry et Shephard (1984). La méthode n'avait pas connu beaucoup de succès à l'époque à cause du fait qu'elle était coûteuse en terme d'espace mémoire et avait même été surnommée méthode "explosive". Mais la conjugaison de l'accroissement de la puissance des ordinateurs, de la baisse continue du prix de la mémoire et d'un choix approprié des

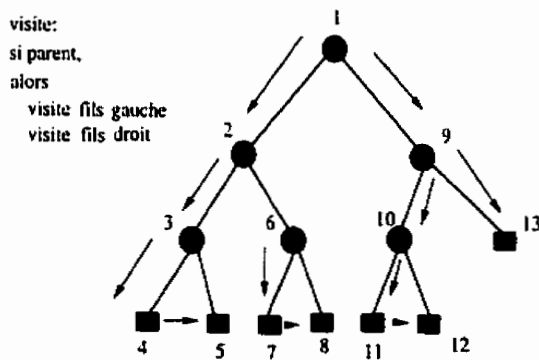
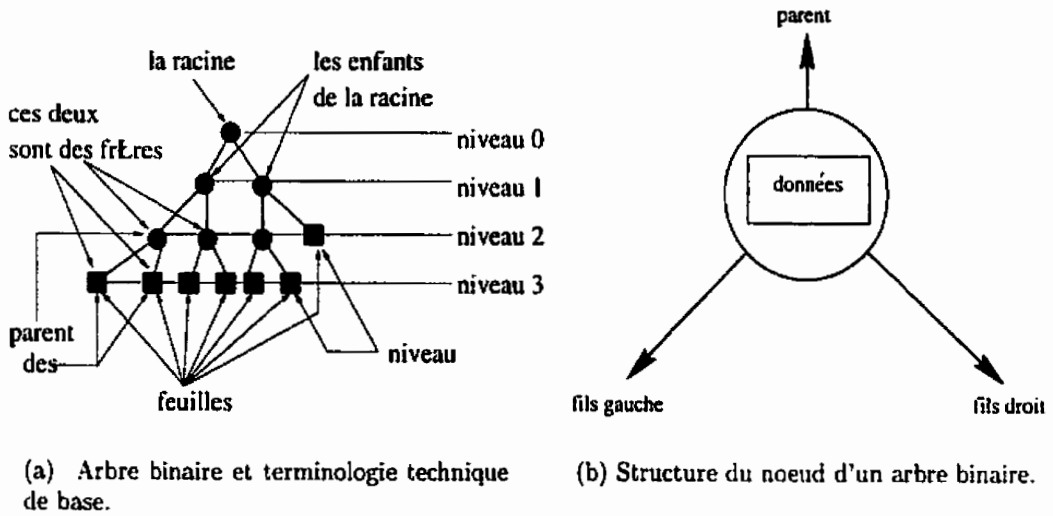


Figure 6.10: Structures, terminologie et opérations de base sur un arbre binaire.

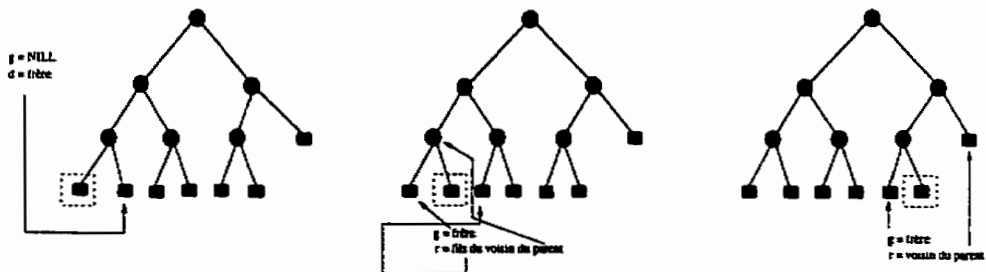


Figure 6.11: Trois possibilités de calcul de voisinage et d'ancestrale.

structures de données fait que cette méthode a suscité au cours des dernières années un regain d'intérêt. Dans les années 1990, Shephard et George (1991), Schroeder et Shephard (1991), Bern et Eppstein (1992), Bern et al. (1990) ont présenté différents types d'algorithmes basés sur le principe de la décomposition spatiale récursive. En 2D, l'algorithme de Bern-Eppstein-Gilbert est automatique et vérifie les propriétés physiques, topologiques et géométriques. Mitchell et Vavasis (1992) ont produit dans un contexte de géométrie numérique (*computational geometry*) l'un des premiers résultats de maillage volumétrique 3D avec démonstration de la validité des critères topologiques, physiques et géométriques tels que vus dans la section (3.3). Le principe simplifié de la méthode en 3D peut se résumer de la manière suivante: l'objet à décomposer est d'abord enfermé dans un cube ayant comme arête le plus grand côté de l'objet. Ce cube est subdivisé en huit cubes identiques suivant l'algorithme ci-dessous:

Algorithme : `Tester_subdiviser(octant/quadrant, niveau)`

Début

Si `Max_paramètre_de_contrôle > niveau` alors

`diviser_octant/quadrant_en huit/quatre`

`compléter_voisinage`

 Pour `i = 1 à 8 (ou 4)` faire

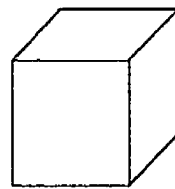
`tester_subdiviser_octant(oct(i), niveau+1)`

 Finfaire

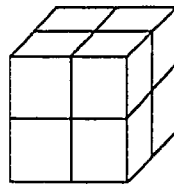
 Finsi

Fin.

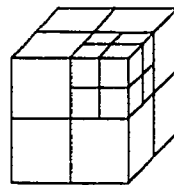
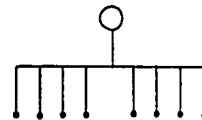
Le cube d'origine sera appelé octant racine, et chacun de ses fils est appelé octant. On teste chaque fils pour voir s'il est à l'intérieur de l'objet (octant intérieur), ou à l'extérieur (octant extérieur), ou bien s'il est partiellement à l'intérieur ou partiellement à l'extérieur (octant non-homogène). Seuls ces derniers seront de nouveaux subdivisés en huit sous-octants comme nous pouvons le visualiser sur les figures (6.12) et (4.4). Le processus précédent est répété jusqu'à l'obtention du niveau de décomposition désiré. Nous obtenons un arbre octree ayant pour noeuds les octants partiels et pour feuilles les octants extérieurs et les octants coupés. Le schéma obtenu par cette méthode peut présenter les défauts suivants si utilisé directement pour générer un maillage :



Cube racine



Niveau 1



Niveau 2

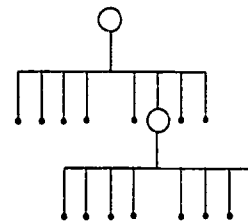


Figure 6.12: Structure Octree.

- L'intérieur du domaine pourrait être représenté par un petit nombre de gros éléments.
- Des octants voisins peuvent se trouver à des niveaux de subdivision différents.
- La représentation de la frontière réelle est mauvaise. Dans ces zones, la frontière sera représentée par des marches d'escaliers susceptibles de fausser les résultats de l'analyse.

Les deux premiers points peuvent être réglés en imposant des contraintes sur les niveaux minimaux de différence de maillage entre octants/quadrants voisins. On crée ainsi des éléments capables d'assurer une transition lisse et douce entre éléments de niveaux de subdivision différents, en effectuant au besoin des subdivisions supplémentaires. On parle alors d'un octree/quadtrees équilibré, sujet abordé

par plusieurs auteurs du domaine dont Bern et al. (1990), Zeeuw (1993), Schneider et al. (1992), Sharifi (1996), Schneider (1996) et que nous visualisons en 2D et 3D par l'intermédiaire de la figure (6.13).

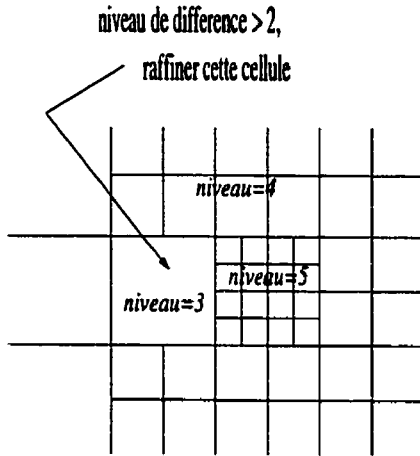
L'algorithme ci-dessous présente les étapes principales d'une subdivision avec un seul niveau de différence :

1. Construire la liste chaînée des Octants/Quadrant feuilles de l'arbre.
2. Trouver les voisinages de chaque Octant/quadrant.
3. Pour tous les Octants/Quadrants de la liste chaînée,
 - (a) Si le niveau de subdivision de l'octant / quadrant est plus petit que le niveau des Octants/Quadrant voisins plus un alors:
 - i. Subdiviser l'octant/quadrant en huit ou en quatre
 - ii. Modifier la liste chaînée des octants/quadrant feuilles
 - A. enlever l'octant quadrant père
 - B. ajouter les octants quadrants fils dans la liste chaînée
 - iii. Mise à jour du voisinage

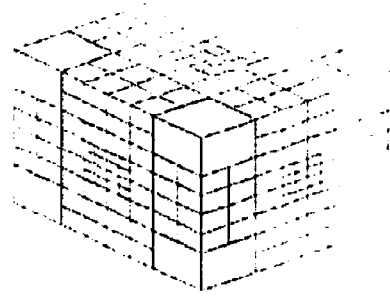
Le troisième point nécessite un traitement complet des éléments de frontière qui peut consister à générer des octants frontières entre les faces libres des octants intérieurs et les surfaces frontières.

6.2.3 Synthèse de la génération de maillage par Octree/Quadtrees

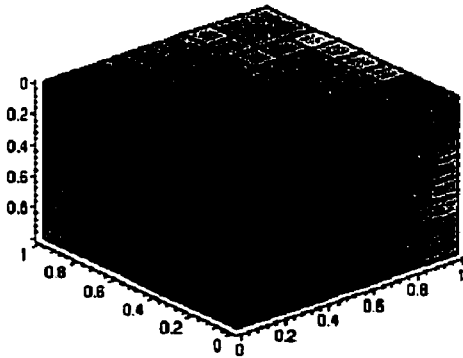
Dans cette approche, on représente le domaine de l'objet par un ensemble de carrés ou cubes disjoints appelés respectivement quatrants et octants qui font partie d'un arbre. A partir de la représentation quadtree/octree, ces auteurs réalisent des maillages utilisables pour la résolution numérique d'équations aux dérivées partielles, en utilisant diverses techniques de contrôles de maillage. D'une manière



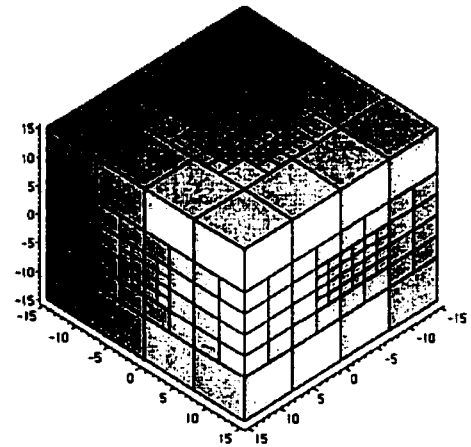
(a) Illustration du rapport 2:1 en 2D.



(b) Illustration du rapport 2:1 en 3D.



(c) Autre illustration du rapport 2:1 en 3D.



(d) Non-respect du rapport 2:1 en 3D.

Figure 6.13: Illustration générale de l'imposition de la différence d'un seul niveau de subdivision, le rapport 2:1.

générale on débute le processus à partir du maillage frontière et d'un certain nombre de paramètres de contrôle du maillage comme la dimension (maximum ou minimum) des éléments, le nombre de niveaux permis dans l'arbre etc. De Rheinbolt et Mesztenyi (1980), Yerry et Shephard (1984), Shephard et George (1991), Schroeder et Shephard (1991), Schneider (1996), Tchon et al. (1997) nous effectuons la synthèse

ci-dessous qui donne les étapes de l'utilisation de la méthode de la décomposition spatiale pour générer un maillage, avec une visualisation partielle par l'intermédiaire des figures (6.12) et (4.4).

1. Entourer le domaine d'intérêt d'un cube (resp. carré) appelé espace universel ou cube mère.
2. Subdiviser le cube mère en 8 octants (carré en 4 quadrants).
 - (a) Tester pour vérifier si les octants (quadrants) sont:
 - i. Pleins, et leurs étiquetages en F.
 - ii. Vides, et leurs étiquetages en V.
 - iii. Partiels, et leurs étiquetages en P.
 - (b) Les octants (quadrants) vides sont déclarés branches terminales et sont enlevés.
 - (c) Les octants/quadrants pleins ou partiels sont testés pour voir s'ils peuvent être subdivisés davantage en se basant sur les informations de contrôle du maillage prédéfinies propres à cette région de l'espace occupé par l'octant/quadrant.
 - i. Si la subdivision est permise, l'octant/quadrant est déclaré inconnu et son noeud est étiqueté noeud de continuation et l'octant est subdivisé de manière récursive.
 - ii. Octants/Quadrants partiels restants à la fin du processus (lorsque le niveau maximum est atteint) sont déclarés octants non-homogènes ou octants coupés.
 - (d) À cette étape, la taille des Octants/Quadrants du modèle est quelconque et englobe encore une partie du domaine extérieur à l'objet considéré et le modèle est dit grossier.
3. Classification des octants dans le but de donner une définition exacte de la topologie des entités frontières issues de l'étape précédente. On réalise une classification des Octants/Quadrants frontières en :
 - (a) Les octants frontières sont classifiés en:

- i. Octants coupés de type face (octant surfacique) si l'octant coupe une ou plusieurs faces.
 - ii. Octants coupés de type arête, (octant arête) si une ou plusieurs arêtes rentrantes ne s'intersectent pas et si un minimum de deux segments de faces y sont reliés.
 - iii. Octants coupés de type sommet (octant sommet) si au moins un sommet avec un minimum de trois segments de face existe dans l'octant.
- 4. Insertion de l'information concernant les octants coupés dans l'arbre en commençant par les sommets, ensuite les arêtes et enfin les faces. Compléter ensuite l'intérieur de l'octree par une traversée de l'arbre en exigeant éventuellement que les Octant/Quadrants partageant ou ayant des arêtes et faces en commun, aient une différence de niveau d'au plus un.
- 5. Génération de maillage dans la méthode octree/quadtrees :
 - (a) Définition de la topologie du maillage sur une base octant (quadrant) par octant (quadrant).
 - i. Utiliser un algorithme de triangulation des octants/quadrants.
 - ii. Utiliser un schéma de maillage prédéfini ou patron de maillage (templates).
 - (b) Finalisation du maillage en trois étapes.
 - i. Positionnement des noeuds en utilisant des informations de connectivité.
 - ii. Redéfinition de la topologie.
 - iii. Repositionnement des noeuds.
 - (c) Utilisation des techniques de contrôle de maillage a priori ou a posteriori pour améliorer la taille, la finesse, ou la densité du maillage.
 - i. Contrôle de la grandeur (taille) des octants/quadrants.
 - ii. Utilisation des tolérances variables associées avec différentes entités géométriques.
 - iii. Utilisation des points de raffinement dans l'espace c'est-à-dire des données de localisation des points dans l'espace où la tolérance doit être modifiée d'une quantité connue.

Chapitre 7

Mise en oeuvre informatique

En partant des problèmes que nous avons soulevés et des objectifs que nous voulons atteindre (production d'un maillage hybride en 3-D), de la synthèse générale de la méthode des octree de la section (6.2.2), nous avons développé un système informatique (le programme Octree) mettant en oeuvre ces idées.

7.1 Bilan de conception du programme Octree

Le programme Octree prend la structure discrète d'un corps tridimensionnel et s'en sert pour générer un maillage composé d'un maillage de peau et d'un maillage espace. Cet ensemble hybride remplit l'espace autour du corps, généralisant ainsi pour la première fois les travaux de Smith et Leschziner. Dans cette section nous présentons d'abord l'algorithme général du programme Octree. Nous présentons ensuite deux diagrammes de flot de données de Octree et enfin l'arbre de construction et l'organigramme des fonctions génériques. Dans la section (A.2), nous expliquons en détail l'obtention de cette structure discrète à partir des routines dont nous disposons dans MapleV.

7.1.1 Les grandes étapes du programme

L'obtention d'un maillage hybride ne peut pas se faire d'un seul coup, mais nécessite qu'on exécute une série d'étapes que l'on peut ramener à l'algorithme général suivant:

1. Modélisation géométrique, discrétisation et récupération des données via MapleV.
2. Génération de l'arbre octree.
3. Traitement de l'octree.
 - (a) Allègement.
 - (b) Imposition de la différence d'un seul niveau de subdivision (voir figure (6.13)).
4. Génération du maillage de peau.
 - (a) Génération et indexation des faces des octants frontières en face de l'objet.
 - (b) Lissage des faces.
 - (c) Projection des noeuds des faces des octants frontières en faces de l'objet.
 - (d) Génération du maillage de peau et du maillage extérieur.
5. Fichier de sortie d'indexation générale et visualisation.

7.1.2 Diagrammes de flots de données du programme

Dans ce type de représentations on identifie les fonctions ou les grandes fonctionnalités d'un programme que l'on relie par la suite par des données. Dans cette optique, les cercles représentent les fonctions et les carrés représentent toutes les données extérieurs ou les sorties du programme. Il existe plusieurs niveaux de détails possible dans un diagramme de flot de données mais à partir d'un certain niveau le DFD (Data Flow Diagram) ressemble à un algorithme, ce qui n'est pas le but visé et un certain pragmatisme s'impose. Pour notre part nous allons décrire le programme Octree à un niveau DFD zéro et à un niveau DFD 1, deux niveaux de détails que l'on peut visualiser par l'intermédiaire des figures (7.1) et (7.2).

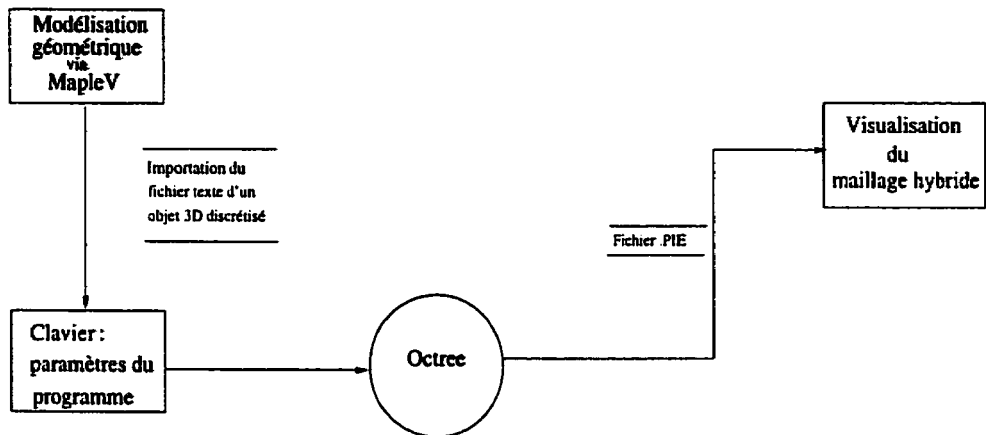


Figure 7.1: Diagramme de flots de données de niveau 0 du programme Octree.

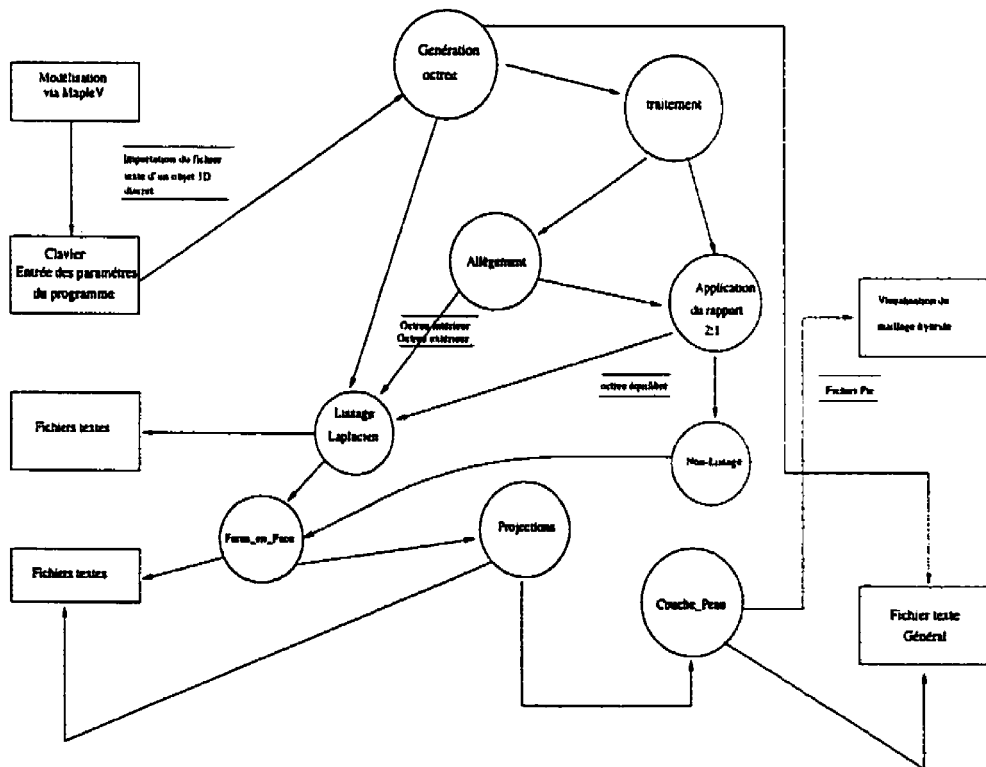


Figure 7.2: Diagramme de flots de données de niveau 1 du programme Octree.

7.1.3 Arbre de construction des modules du programme

Pour implanter l'algorithme précédent, le programme Octree a été structuré d'une façon modulaire. Le schéma global est illustré à la figure (7.3). Cette figure

présente l'arbre de construction du programme. Le programme est composé de quatre modules principaux: initialisation et récupération des données, génération octree, opérations numériques globales et le module de sortie et maillages hybrides. Chaque module peut aussi utiliser d'autres petits modules et encapsule les procédures et les données nécessaires pour faire son travail.

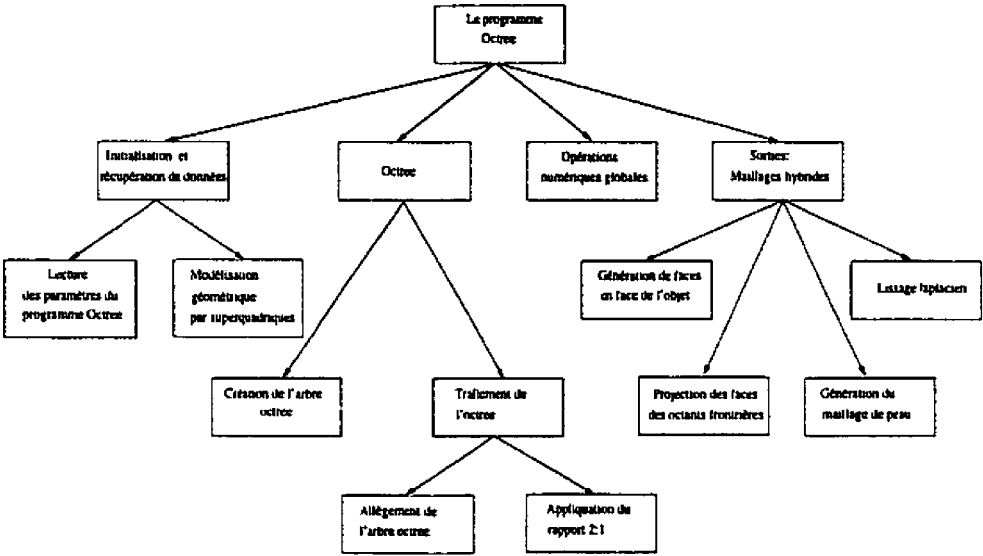


Figure 7.3: Arbre de construction du programme Octree.

7.1.4 Digramme hiérarchique des fonctions génériques du programme

Pour une autre vision du programme Octree observons les figures (7.4) et (7.5) en partant de son point d'entrée. On ne traite ici que des fonctions génériques importantes: les fonctions rattachées aux deux classes Array et V3D développées pour Octree sont en effet d'usage tellement courant qu'elles n'auraient fait qu'alourdir cette section. Elles seront cependant présentées dans la section suivante.

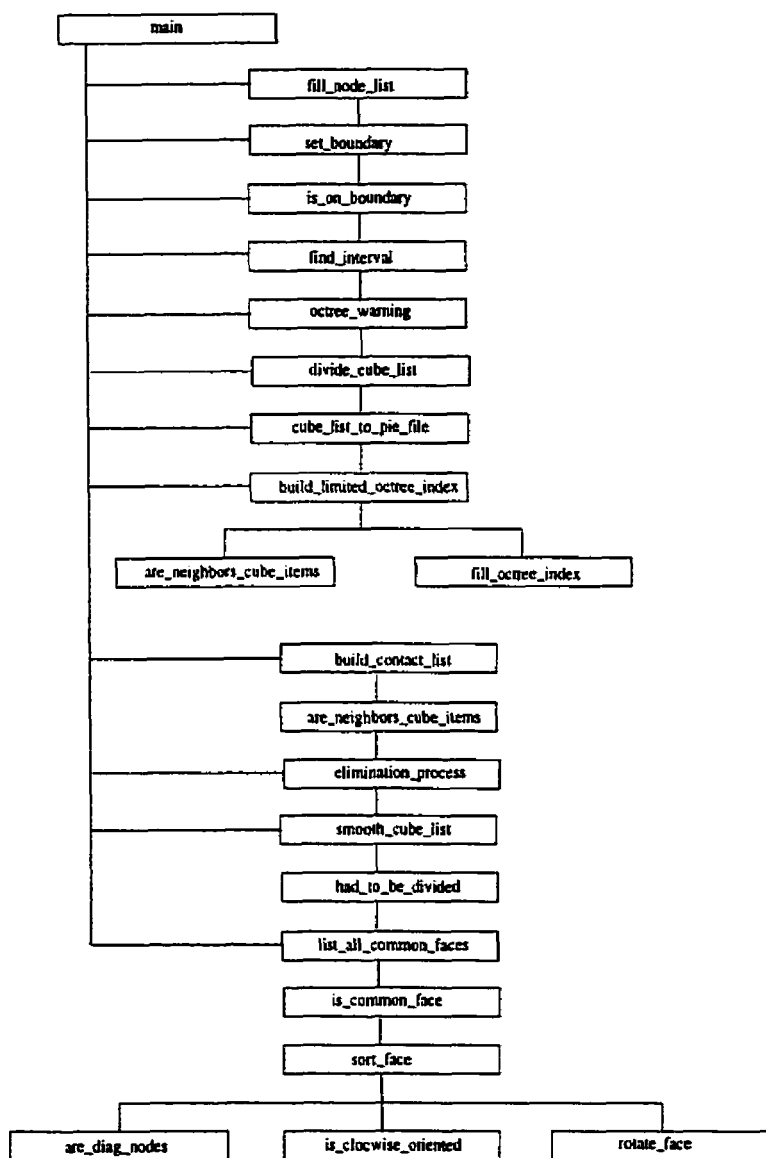


Figure 7.4: Arborescence des fonctions génériques de Octree.

7.2 Design conceptuel, mise en oeuvre informatique et évaluation des algorithmes

Le passage pour la première fois en 3D de la méthode de Smith et Leschziner a exigé des contributions au niveau de la structuration des données et des contributions au niveau algorithmique. Dans cette section nous présentons de nouveaux

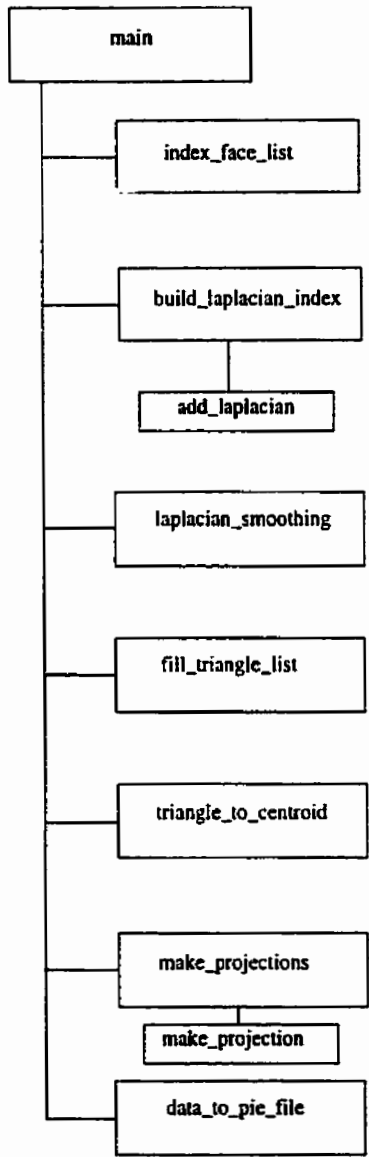


Figure 7.5: Suite de l'arborescence des fonctions génériques.

algorithmes de calculs d'intersection, d'ordonnancement des faces, de projection et de génération de couches structurées de peau.

7.2.1 Détection et calcul d'intersection

Le calcul des intersections est un aspect fondamental dans la méthode de la décomposition spatiale, car c'est la présence ou non de l'objet ou d'une partie de l'objet dans un octant qui va déterminer si l'octant concerné va être subséquentement subdivisé ou pas. l'algorithme suivant que nous proposons repose sur une gestion de deux listes.

liste 1 - Octants sur la frontière

liste 2 - Octants qui ne sont pas sur la frontière

- Génération de l'octant de départ, jugé en dehors de la frontière (liste 2)

- POUR k fois (k est le niveau de subdivision fixé au départ)

- POUR chaque octant dans la liste 2

- SI l'octant est sur la frontière ALORS

-Transférer de la liste 2 à 1

SINON

- Diviser en huit dans la liste 1

Une évaluation de cet algorithme permet de voir qu'il y a deux boucles imbriquées. Si on suppose qu'il y a n octants sur la frontière et k niveaux de subdivision, l'algorithme sera de l'ordre de $O(kn)$. Les opérations de passage d'une liste à une autre sont des tests d'affectations, donc s'effectuent en temps constant.

7.2.2 Indexation générale des faces

On remarquera d'abord qu'un octant possède un maximum de 26 voisins: 6 voisins de face, 12 voisins d'arête, 8 voisins de coins. Il est donc important et fondamental d'ordonner les noeuds, faces, et octants d'une manière unique avant toute autre opération et c'est un problème informatiquement difficile et délicat. La routine qui s'occupe de cette mise en ordre s'appelle *sort_face*. En entrée de la fonction on fournit une structure de type face c'est-à-dire:

```

struct Face {
    double x1, y1, z1,
           x2, y2, z2,
           x3, y3, z3,
           x4, y4, z4;
};

```

qui donne les coordonnées cartésiennes des quatre noeuds d'une face rectangulaire, dans n'importe quel des $4!$ ordres possibles. La face est considérée plane et normale à i , j , ou k et en sortie de la fonction on a:

- Les noeuds sont consécutifs, le i ème noeud listé est voisin (n'est pas en diagonale) du $i+1$ ème noeud, même chose pour le dernier et le premier noeud.
- Les noeuds sont positivement orientés par rapport à l'origine: en appelant $n1$ le premier noeud, $n2$ le second, etc, les noeuds sont orientés de façon telle que le produit scalaire de $O.n1$ et de $n1.n2 \times n2.n3$ est positif ou nul, O étant l'origine.
- Le premier noeud est "invariant". Pour une liste de quatre noeuds donnés, une recette interne est implémentée qui fait qu'un noeud des quatres noeuds est systématiquement choisi pour être le premier.

Cette façon de faire permet de sélectionner de façon invariante parmi 24 tris possibles une seule et même façon de décrire une face. C'est une caractéristique utile qui permet de comparer deux faces données par une opération bit à bit. De là on peut bâtir le fichier de sortie d'indexation générale qui contient pour une simulation donnée les informations et relations suivantes:

- Les informations générales sur les octants (nombre total d'octants, nombre d'octants sur la frontière de l'objet, nombre d'octants qui ne sont pas sur la frontière);
- Les numéros de relation de voisinage octant-octant;
- Les relations volume-face de chaque octant;

- Les coordonnées (x,y,z) des centroïdes des de chaque octant;
- Les relations face-sommet;
- Les numéros des faces des octants frontières en face avec l'objet (fondamentale pour la projection et la génération du maillage de peau, (voir illustration à la figure (7.6)));
- Les coordonnées (x, y, z) des centroïdes faces de chaque octant;

7.2.3 La projection

C'est un aspect fondamental de la génération du maillage de peau par la méthode de Smith et Leschziner. C'est cette opération qui permet d'insérer les couches d'éléments structurés formant le maillage de peau dans l'espace libre entre l'objet et le maillage extérieur. La démarche proposée ici s'effectue en trois phases: avant, pendant et après la projection. Dans la fonction d'entrée principale du programme, une fois que l'octree est généré, traité et que les faces sont elles-mêmes créées, indexées et lissées, on obtient les structures suivantes qui décrivent les faces des octants frontières en face de l'objet:

- *Array < Node > node_list*. Les coordonnées, sans répétitions, de l'ensemble des noeuds des faces en face de l'objet.
- *Array < Face > boundary_face_list*. De chaque élément de cette structure, seuls les champs $x1, x2, x3, x4$ sont significatifs: leurs valeurs entières (il s'agit de doubles) contiennent la valeur d'index des sommets de *node_list* qui composent une face initialement rectangulaire. On remarquera que la face n'est plus forcément rectangulaire, puisque la position des sommets a été lissée. De la face rectangulaire $x1 - x2 - x3 - x4$, on peut cependant tirer deux sous-faces planes triangulaires $x1 - x2 - x3$ et $x1 - x3 - x4$, par exemple.

7.2.3.1 Étapes préliminaires à la projection

À partir de l'étape de la section (7.2.3), on recharge le maillage triangulaire qui représente l'objet original que l'on stocke dans *Array < Triangle > triangle_list*.

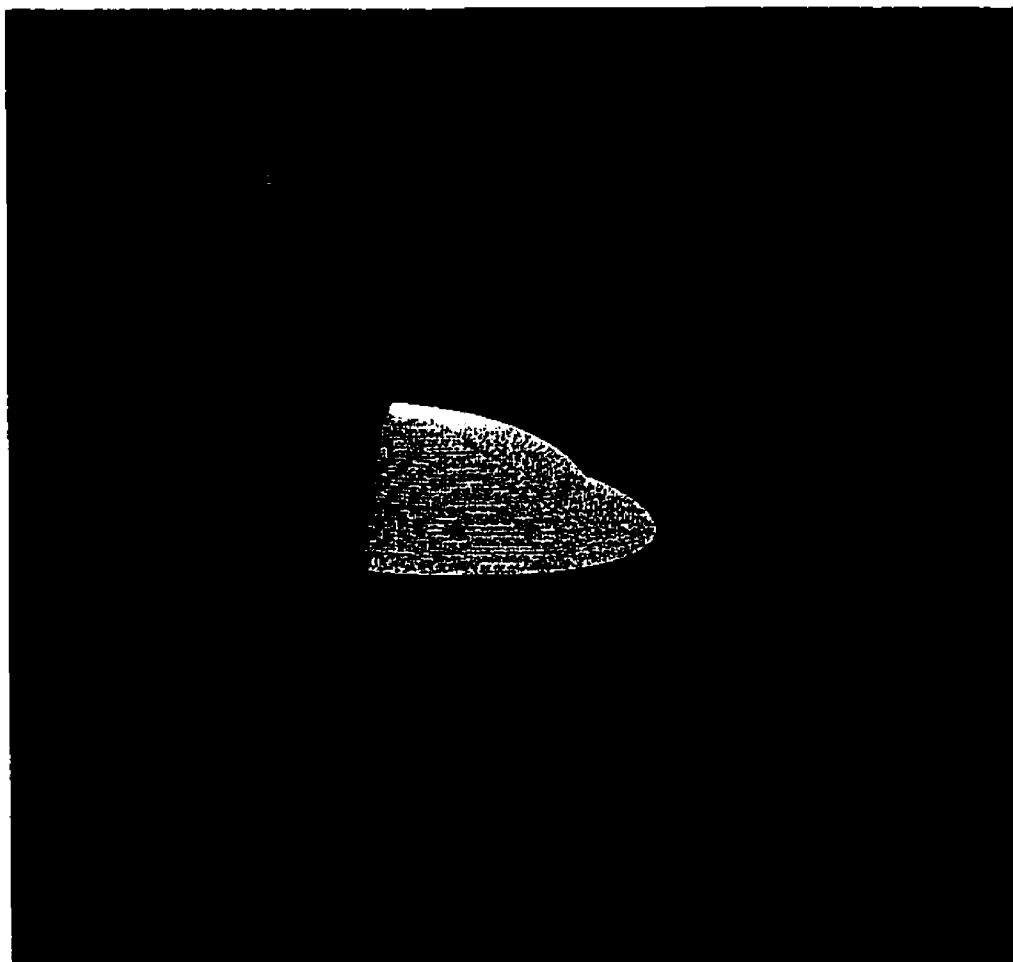


Figure 7.6: Une visualisation des octants frontières en face d'un objet.

On crée ensuite une table de correspondances contenant l'ensemble des centroïdes des faces triangulaires du maillage précédent. Le calcul des centroïdes se ramène aux calcul des points dont les coordonnées sont les moyennes des coordonnées correspondantes des trois sommets de chaque face. La projection à proprement parler, se fait en deux étapes: la recherche de la meilleure face triangulaire de l'objet sur laquelle projeter et ensuite on procède à la projection en tant que telle.

7.2.3.2 Première étape de la projection: La fonction *make_projections*

On transmet à la fonction *make_projections*, les structures *node_list*, *triangle_list*, *centroid_index*, précédemment décrites. La tâche centrale de

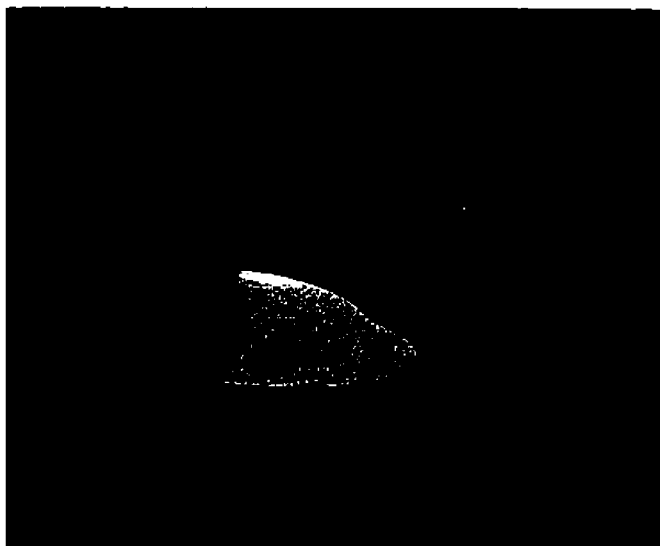


Figure 7.7: Une visualisation des faces des octants frontières en face d'un objet: cas du nez de l'avion.

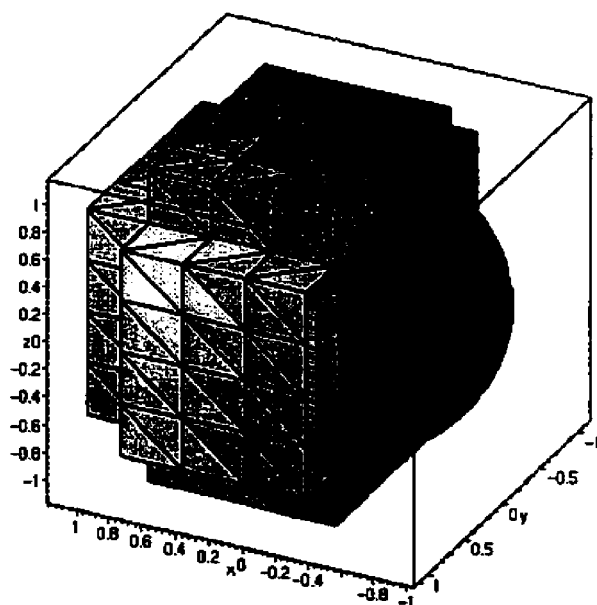


Figure 7.8: Une visualisation des faces des octants frontières en face d'un objet: cas d'une sphère.

make_projections est de trouver la meilleure face sur laquelle projeter pour chacun des noeuds de *node_list*. L'idée derrière cet algorithme est la suivante: pour des points suffisamment proches de l'objet (tels ceux qu'on traite) la face la plus intéressante est la face la plus proche du point. On détermine d'abord sans projection, une bonne approximation de cette *face la plus proche* en choisissant de projeter le point sur la face dont le centroïde est à la distance la plus faible. L'algorithme se présente sous la forme suivante:

POUR chaque noeud de la liste

 Trouver le centroïde le plus proche

 Projeter sur la face qui a pour centroïde celui le plus proche

FIN

En Pseudo code nous avons:

```
make_projections(node_list, triangle_list, centroid_index)
    int i, j, i_max, j_max, best_index;
    double    current_distance, best_distance;
    V3D        current_vertex;
    V3D        projection_index;
    Node        data_1;
    Triangle    data_2;
    Pour (i=0; i<i_max; ++i) {
        best_distance=1E10;
        current_vertex.set(data_1[i]->x, data_1[i]->y, data_1[i]->z);
        Pour (j=0; j<j_max; ++j) {
            current_distance= (current_vertex-centroid_index[j]);
            Si (current_distance<best_distance);
                best_distance=current_distance;
                best_index=j;
        }
        projection_index[i]=make_projection(best_index, current_vertex);
    }
    return(projection_index);
```

7.2.3.3 Deuxième étape: la fonction make-projection

Le dernier point de la projection est réalisée par cette sous-fonction, à laquelle on transmet:

- Les coordonnées des sommets de la face triangulaire sur laquelle projeter.
- Les coordonnées du point *p* à projeter.

Le travail de projection en tant que tel s'effectue de la manière suivante: on déplace le point suivant le vecteur résultant de la projection orthogonale sur la normale au plan qui sous-tend la face triangulaire d'un vecteur quelconque. Partant d'un point *p* à un point quelconque du plan qui peut être un sommet par exemple. Les choses se compliquent par la suite, quand il s'agit de déterminer si le point projeté appelé *p_prime*, est à l'intérieur de la section triangulaire du plan qui compose la face. Le procédé utilisé, bien que géométriquement clair, peut être difficile à verbaliser: Contentons-nous d'un résumé opérationnel. On commence par créer, à partir des sommets *p1*, *p2*, *p3* et de *p_prime*, six vecteurs: *p1-p_prime*, *p2-p_prime*, *p3-p_prime*, *p1-p2*, *p2-p3*, *p3-p1*. Par produit vectoriel on crée trois vecteurs qui sont:

```
cross_set_1 = p1-p2 X p1-p_prime
cross_set_2 = p2-p3 X p2-p_prime
cross_set_3 = p3-p1 X p3-p_prime
```

Si les produits scalaires deux à deux des vecteurs *cross_set_1*, *cross_set_2*, et *cross_set_3* sont tous positifs ou nuls, alors le point *p_prime* est à l'intérieur du triangle quelconque ou sur une de ses frontières. Dans le cas inverse, on projette *p_prime* sur le sommet le plus proche suivant l'algorithme ci-dessous:

Projection de *p* sur le plan qui sous-tend la face triangulaire

SI la projection est à l'intérieur de la face

ALORS Ok. Renvoyez ce résultat à la fonction appelante

SINON Renvoyez le sommet de la face la plus proche à la fonction appelante

Fin Si

Au fur et à mesure de l'exécution de *make_projections*, les données sont stockées dans un tableau dynamiquement alloué et sont finalement renvoyées par pointeur à la toute fin dans la structure: *V3D *projection_index* qui contient donc sous forme vectorielle la projection, dans l'ordre de chacun des points des sommets des faces en face de l'objet stockés dans *node_list*.

7.2.3.4 Analyse de la complexité algorithmique de la projection

Le temps de calcul des instructions simples d'assignation, de lecture et de stockage est constant. La fonction *make_projections* fait appel à la fonction *make_projection*. Dans cette sous-fonction qui n'est pas une récursion, on remarque que les *if – then – else* ne font pas appel à leurs tours à des fonctions. On teste tout simplement des conditions et comme conséquence ce sont des opérations qui se font en temps constant. Il reste donc à évaluer le temps de calcul de l'instruction *for*, ce qui revient à trouver le temps de calcul du corps de la boucle. Pour cela il faut répondre à la question du nombre de fois que cette instruction est répétée. On voit que pour n noeuds de faces d'octants frontière en face de l'objet l'opération est réalisée n fois, ce qui donne dans le pire des cas une complexité de l'ordre de $O(n^2)$. L'algorithme est précis et performant à tel point que l'on peut représenter les objets uniquement par la trace des barres de la projection des sommets des faces des octants frontières en face de l'objet sur celui-ci comme illustré et visualisable par les figures (7.11) et (7.12). L'algorithme de projection que nous proposons ici est plus simple, efficace, entièrement automatique par rapport à celui de la technique de projection dans un contexte NURBS de la section (7.2.3).

7.2.4 Génération des couches structurées de peau

Trois structures déjà décrites dans l'étape de projections sont nécessaire à la génération d'éléments prismatiques qui vont venir remplir l'espace vide situé entre les faces des octants frontières en face de l'objet (exemple de la figure 6.13) et l'objet. L'idée à la base de la génération des couches de maillages de peau, découle du format adopté par les fichiers .PIE du logiciel Vu pour décrire un maillage. Une zone d'un maillage, dans son expression la plus simple, y est notée:



Figure 7.9: Représentation octree à deux niveaux de décomposition du nez de l'avion.



Figure 7.10: Un zoom des projetés des noeuds des faces des octants frontières en face de l'objet.

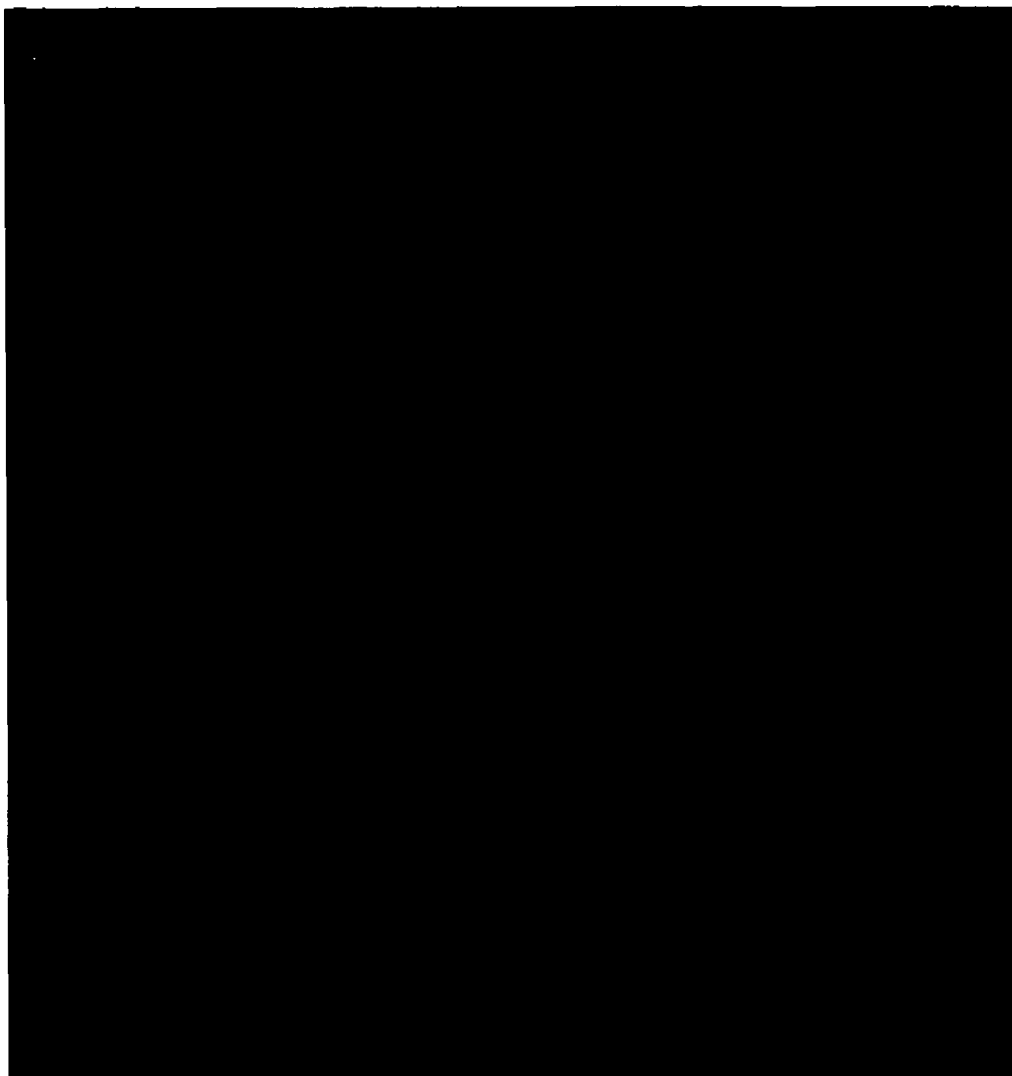


Figure 7.11: Une sphère représentée de manière précise uniquement par la trace de la projection des noeuds des faces des octants frontières en faces de l'objet.

- ZONE my-zone interpolant, coordonnées, connections, *nombre_de_mailles*.
- Interpolant: la nature du maillage à générer, ici des prismes triangulaires à 6 sommets.
- Coordonnées: un champ donnant l'ensemble des coordonnées utilisées sous forme de triplet (x, y, z) .

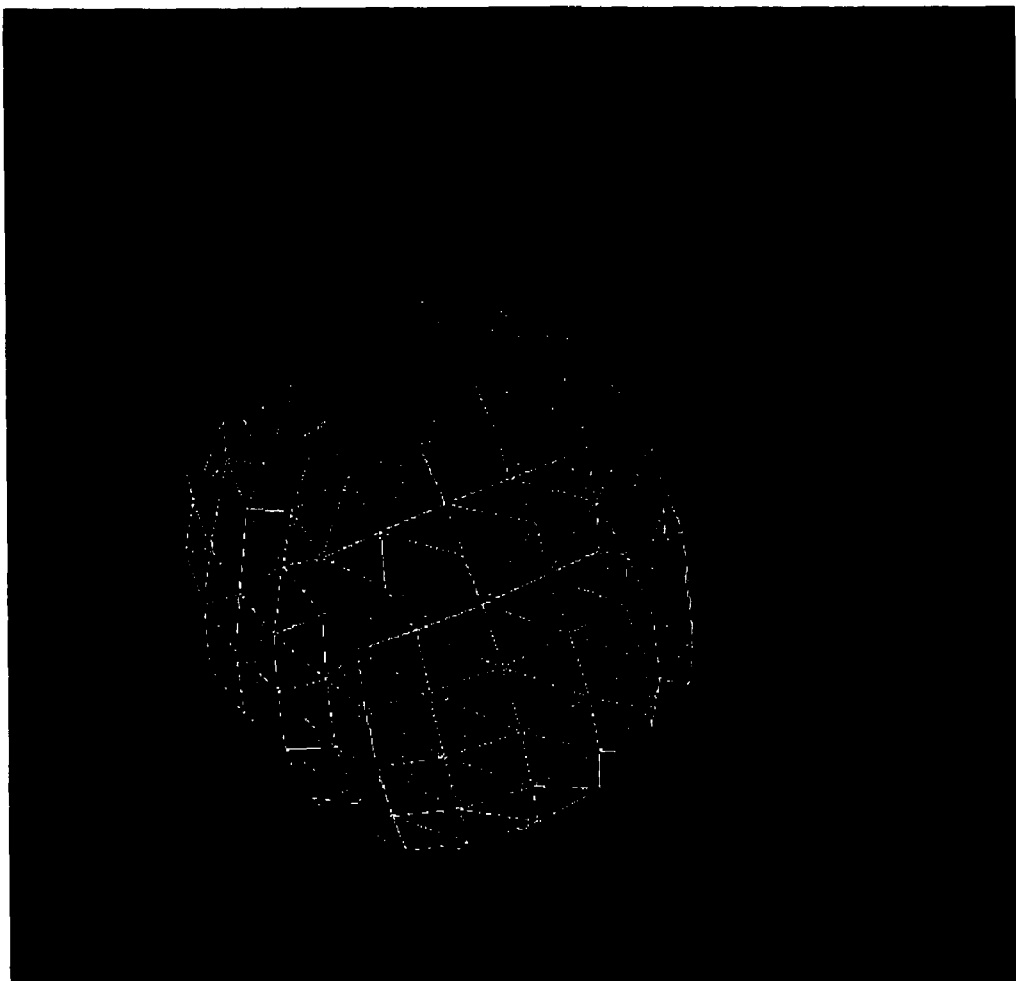


Figure 7.12: Une autre visualisation de la projection des noeuds des faces des octants de la frontière en face de l'objet.

- **Connections:** par groupe de six nombres, les numéros d'index des sommets de la liste précédente à réutiliser pour former chacune des mailles.
- **Nombre-de-mailles:** un scalaire ne donnant rien d'autre que le nombre de groupe de six nombres de la liste précédente.

Nous avons donc obtenu, par les étapes précédentes du programme, un maillage de surface des faces des octants frontières en face de l'objet, stocké dans un format similaire à celui du fichier PIE, les coordonnées d'un côté dans *node.list*, les indexations de l'autre dans *boundary_face.list*. On connaît également les coordonnées des

noeuds de *node_list* une fois projetés sur le corps. Pour obtenir un maillage prismatique séparé en couches distinctes, prises en "sandwich" entre les faces des octants frontières et l'objet lui-même, on effectue une simple "extrusion" de ces faces. Pour cela on reliera chaque noeud de chacune des faces à un autre noeud d'interpolation situé à une certaine proportion du segment reliant ce premier noeud à sa projection connue sur la surface de l'objet et comme illustré par la figure (7.13). Prenons

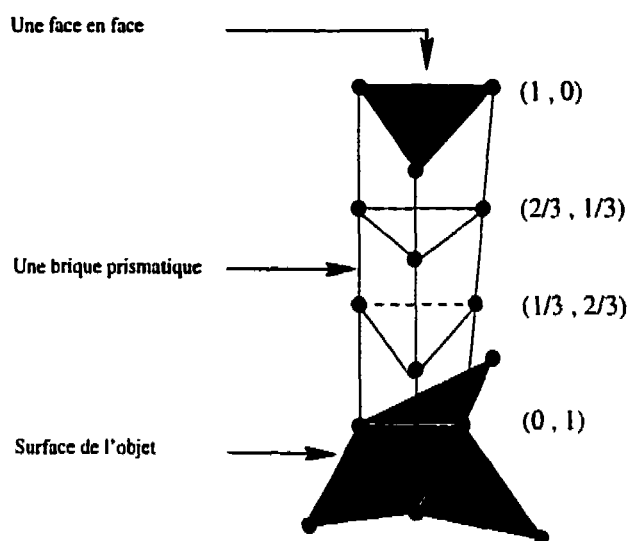


Figure 7.13: Topologie prismatique du maillage de 3 couches de peau.

un autre exemple: imaginons que nous possédions une face triangulaire ainsi que la projection de chacun de ses sommets sur l'objet, et qu'on désire obtenir deux mailles allant dans deux couches de peau distinctes s'emboîtant l'une dans l'autre. Il suffira de calculer, par simple moyenne des coordonnées, trois sommets médians, situés au milieu des segments de droite reliant les sommets de notre face à leurs projections. On créera ensuite une première maille prismatique en utilisant comme sommets ceux de la face et ceux *interpolés*, et une seconde en utilisant ces derniers et les sommets de projection. Dans les faits, imaginons que nous possédions n faces triangulaires (décrites par k sommets distincts, $n \leq k \leq 3 * n$, et k projections correspondantes), et que nous désirions générer i couches de peau. Il nous faudra donc posséder, par voisinage, $(i + 1)$ couches surfaciques donnant les bases et les sommets de nos mailles prismatiques, ceci incluant les faces originales et les faces générées par une connection identique des noeuds de projections correspondants. Il faudra, en clair, générer $(i - 1)$ couches moyennes de noeuds interpolés (simple

processus de moyenne pondérée entre le noeud original et sa projection), qu'on connectera selon le même patron que le patron d'origine des faces. Pour simplifier les choses, toutes les couches de peau (nommées *ZONE skin[x]*) font appel à une même table de coordonnées très imposante (nommée *CoordSkin*), structurée ainsi:

- Triplets 1 à k : Les coordonnées de *node.list*.
- Triplets $k+1$ à $2*k$: Première couche des coordonnées d'interpolation, faisant une bijection avec les autres.
- Triplets $2*k+1$ à $3*k$: Deuxième couche des coordonnées d'interpolation etc.
- Triplets $2*(i+1)+1$ à $2*(i+2)$: Les coordonnées des projections des sommets de *node.list*.

Chaque couche de peau utilise un champ de connection (appelé *SkinConnec[x]*) différent. Pour la structure des "*n* hexlets" qui seront donnés pour chacune de ces couches à l'intérieur de *SkinConnec*, il s'agira de connecter la base et le sommet de chacune des mailles hexagonales en utilisant exactement le même patron que celui de la structure *boundary_face.list*. Supposons qu'on désire créer les mailles hexagonales de deux couches de peau à partir des noeuds 2, 4 et 5, qui forment une face d'après *boundary_face.list*, qui contient la description de 100 faces. En créant un champ *CoordSkin* tel que décrit ci-haut, on peut directement écrire: 2, 4, 5, 2+100, 4+100, 5+100 donnent les noeuds de la première maille. 2+100, 4+100, 5+100, 2+200, 4+200, 5+200 donnent les noeuds de la seconde maille.

7.2.4.1 Structures de données utiles à l'opération

- *Array < Node > node.list*. Les coordonnées, sans répétitions, de l'ensemble des noeuds des faces en face de l'objet.
- *Array < Face > boundary_face.list*. De chaque élément, seuls les champs x_1, x_2, x_3, x_4 sont significatifs: leurs valeurs entières (il s'agit de doubles) contiennent la valeur d'index des sommets de *node.list* qui composent une face initialement rectangulaire. La face n'est plus forcément rectangulaire, puisque la position des sommets a été lissée.

- *V3D * projection.index*. Un tableau dynamique contenant des instances de la classe V3D donnant les coordonnées des projections sur l'objet de chacun des noeuds, dans le même ordre, contenus dans *node.list*. En plus de ces trois structures, un paramètre, *skin.layer.number*, est passé à la procédure *data.to.pie.file*. Il indique le nombre de couches de peau à générer.

7.3 Les structures de données du programme

7.3.1 Les classes Array et V3D

Beaucoup des opérations de Octree sont très routinières: stockage en mémoire, classement de structures de toute nature, et algèbre linéaire dans R^3 . Ces deux catégories, par souci de simplicité et d'efficacité, sont entièrement supportées respectivement par les classes Array et V3D. Fondamentalement il s'agit d'un patron générique de classes qui fournit un ensemble de méthodes permettant la manipulation en mémoire de virtuellement tout ensemble de n données dont chacune d'entre elles peut être stockée statiquement dans un champ de type `struct...{...}`. Cette structure fut fabriquée dans le but de conserver une bonne vitesse d'exécution en manipulant des volumes importants de données, la technique de décomposition spatiale peut devenir très lourde: Des tests préliminaires sous MapleV ont clairement indiqué que la construction d'un octree par des méthodes inappropriées peut être informatiquement explosive. Avec la structure Array nous favorisons la vitesse au détriment de la compacité en mémoire. Selon la nature du traitement envisagé, la structure Array est capable de stocker les données de deux façons différentes: par défaut, dans un mode flottant *FLOATING_ARRAY* ou bien dans un mode statique *FIXED_ARRAY*. Le mode flottant consiste en une liste chaînée à lien double. Il existe une série de routines qui permettent de manipuler une liste dans ce mode, et de naviguer à travers celle-ci avec un *curseur* sur un *élément courant*. C'est également dans ce mode que l'ajout, la suppression, la recherche, les combinaisons d'éléments sont possibles. Pour sa part, le mode statique génère un tableau de pointeurs qui pointent sur une liste d'éléments et dont l'élément terminal est un pointeur NULL. Ce mode est très efficace quand il s'agit d'exécuter des recherches séquentielles, ou encore d'effectuer des tris. L'algorithme Quicksort, un algorithme

inventé par C.A.R. Hoare en 1960 de complexité $n \log(n)$ dans le meilleur cas et n^2 dans le pire cas (Sedgewick (1992)) y a d'ailleurs été implémenté. Ce choix répond aux objectifs que nous nous sommes fixés depuis la sous-section (2.2). Il faut également noter qu'en tout temps on peut changer le statut d'une instance de la classe *Array* en utilisant la méthode *set_status* et ainsi passer d'un mode à l'autre, la classe s'occupant elle-même de la gestion dynamique de la mémoire.

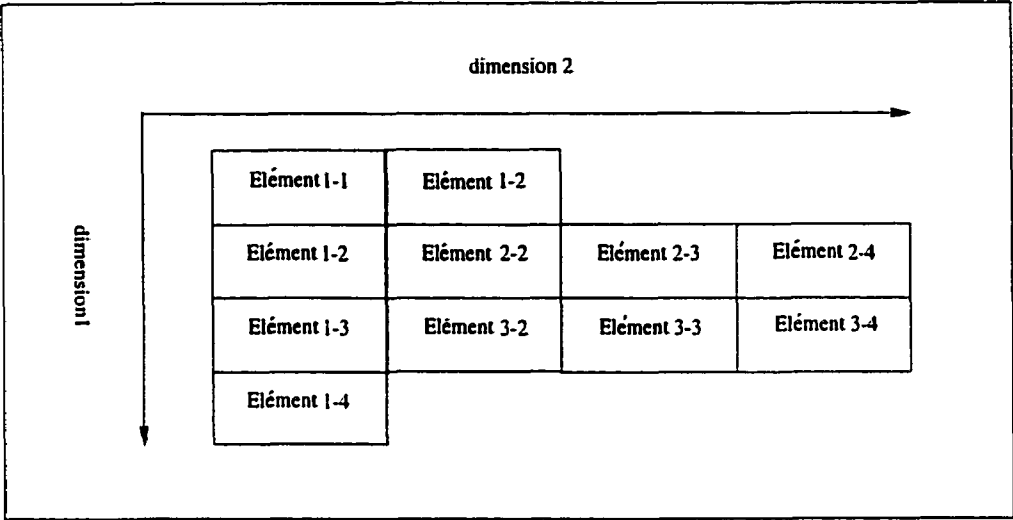
7.3.2 L'algèbre linéaire en dimension trois et la classe *V3D*

La classe *V3D* s'occupe simplement d'alléger les traitements ultérieurs faisant appel à des bases d'algèbre linéaire. Elle permet d'exécuter rapidement sur des vecteurs de V^3 des opérations telles l'addition vectorielle, les produits scalaires et vectoriels, la multiplication scalaire, le calcul de la norme, etc. Pour un maximum de flexibilité, la majorité de ces fonctions sont implémentées sous la forme de surcharges des opérateurs usuels. Par souci de simplicité on considère que la base de l'espace vectoriel utilisé est orthonormale. Au point de vue des propriétés, cette classe ne possède que trois champs, les composantes *i*, *j* et *k* (stockées sur des doubles). On peut utiliser cette classe en conjonction avec la précédente pour certains traitements sous la forme: *Array < V3D > my_V3D_list*;

7.3.3 Règles générales de stockage

Généralement, les structures de données du programme *Octree* sont simples, générales et efficaces. Il s'agit de listes mono dimensionnelles d'éléments de type *struct my_element_struct{...}*. Selon que ces listes étaient appelées ou non à changer en taille durant l'exécution du programme, elles furent stockées respectivement dans une classe *Array < my_element_struct > my_element_list* ou dans un tableau alloué une fois pour toute en mémoire dynamique. Les quelques structures restantes de nature plus complexes, les tableaux bidimensionnels dont la taille dans la seconde dimension varie selon l'élément (tableau 7.1), sont d'usage suffisamment restreint pour qu'on puisse se contenter de les stocker dans des tableaux bidimensionnels initialisés au nombre maximum d'éléments possible selon la seconde dimension, et qu'on s'occupe ensuite d'en gérer les exceptions. On peut faire entrer

Tableau 7.1: Structures de données à seconde dimension variable.



l'ensemble des structures de données utilisées par le programme Octree dans l'une des trois catégories précédentes. Nous illustrons la méthode de stockage choisie par le tableau (7.2) et dans le tableau (7.3) nous faisons une synthèse de toutes les structures importantes et de leur utilisation.

Tableau 7.2: Les structures de données selon leur contenu.

Quantités importantes de données du même type , nombre inconnu de structures à la création invariable durant l'exécution	Utilisation de la classe Array appropriée Array<my_struct>my_struct_list;
Quantités importantes ou non de données du même type, nombre connu de structures à la création invariable durant l'exécution	Utilisation d'un tableau mono-dimensionnel alloué dynamiquement à la création
Données concernant les voisinages	Tableau bidimensionnel alloué dynamiquement

Tableau 7.3: Les structures importantes du programme Octree.

NOM	TYPE	UTILISATION
node_list	Class_Array	Stocke les coordonnées des noeuds du corps d'origine
boundary_cube_list	Class_Array	Stocke les coordonnées des noeuds du corps d'origine
not_boundary_cub_list	Class_Array	Stocke les octants en dehors de la frontière
boundary_face_list	Class_Array	Stocke les faces communes aux deux structures précédentes
triangle_list	Class_Array	Stocke les faces triangulaires composant l'objet d'origine
centroid_index	Tableau mono-dimensionnel	Stocke l'ensemble des centroides des faces de la liste précédente
projection_index	Tableau mono dimensionnel	Projection de tous les noeuds appartenant à triangle_list
laplacian_index	Tableau bidimensionnel	Contient le numéro des noeuds voisins de tous les autres noeuds

Chapitre 8

Quelques exemples d'Applications

Pour démontrer la capacité du programme Octree, nous allons l'appliquer sur quelques objets géométriques qui ont été modélisés et discrétisés par l'intermédiaire du langage MapleV. Le programme Octree que nous proposons est basé sur la technique de décomposition par octree. Cette phase d'application s'effectuera de la manière suivante:

- a Nous partons avec le nez de l'avion modélisé dans la section (6.1.3) et nous expliquons et visualisons en **détail** étape par étape la génération d'une représentation octree avec un seul niveau de subdivision, comment nous effectuons la projection, comment nous générons les couches de peau, etc. Cette étape sera généralisée par une représentation octree complète du nez de l'avion vue sous plusieurs angles.
- b Nous validons ensuite le programme octree sur une sphère, et nous faisons un choix de présentation en minipage ce qui permet en un seul coup d'oeil d'avoir une vision globale et complète des grandes étapes d'une simulation effectuée par le programme Octree.
- c Nous terminons cette phase par des sorties de résultats de simulation du programme Octree appliqué sur quelques objets géométriques composites comme par exemple:
 - (a) objet géométrique composite d'une sphère et d'un cône.

(b) Objet géométrique composite tore-cylindre.

8.0.4 Le programme Octree appliqué sur le nez d'un avion

Pour illustrer le déroulement du programme Octree sur le nez de l'avion modélisé par superellipses, nous partons avec l'objet présenté à la figure (8.1). Cet objet

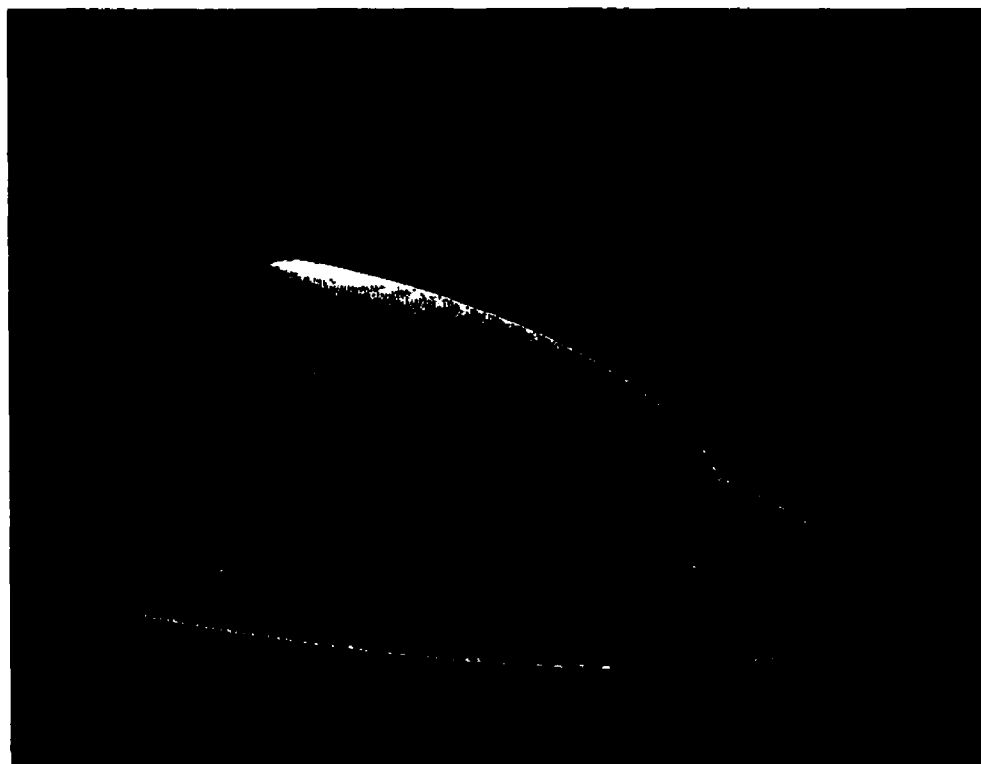


Figure 8.1: Nez avion modélisé par superellipses.

est enfermé dans sa boîte englobante que nous avons volontairement resserrée, ce qui explique un débordement de l'objet à travers la structure octree. On peut aussi remarquer que le contenant de l'objet (sa boîte englobante), a subi une seule itération de décomposition ce qui donne un total de 8 octants (4 octants frontières et 4 octants non-frontières) tel que visualisable sur la figure (8.2). A cette étape on élimine les octants directement en contact avec l'objet ce qui donne la figure (8.3) et on projette sur l'objet les noeuds des faces des octants frontières en face de l'objet. Le maillage de peau sera généré dans l'espace compris entre les faces des

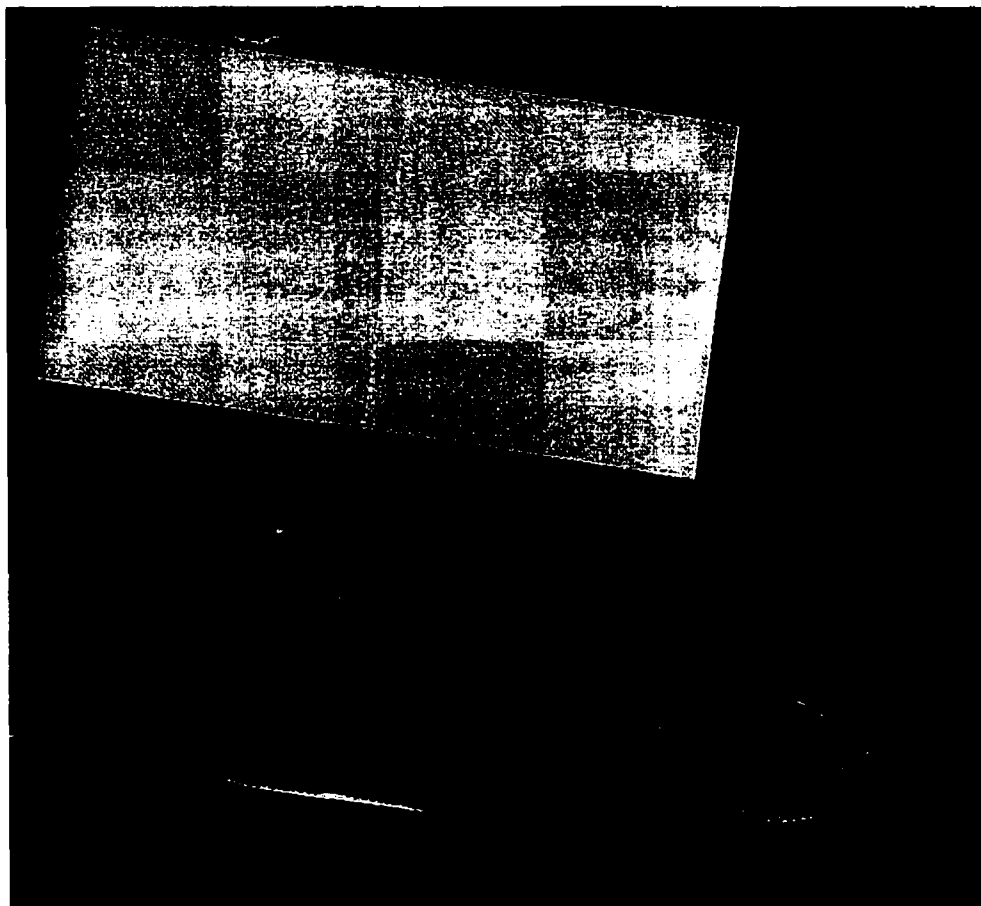


Figure 8.2: Nez avion dans sa boîte englobante en une seule itération du programme Octree. On remarquera 4 octants frontières et 4 octants non-frontières.

octants frontières et l'objet tel que nous l'avons expliqué dans la section (7.2.4). Dans cet exemple nous choisissons de générer 2 couches de peau tel que visualisable sur la figure (8.6) pour la première couche de peau, et sur la figure (8.7) pour la deuxième couche de peau.

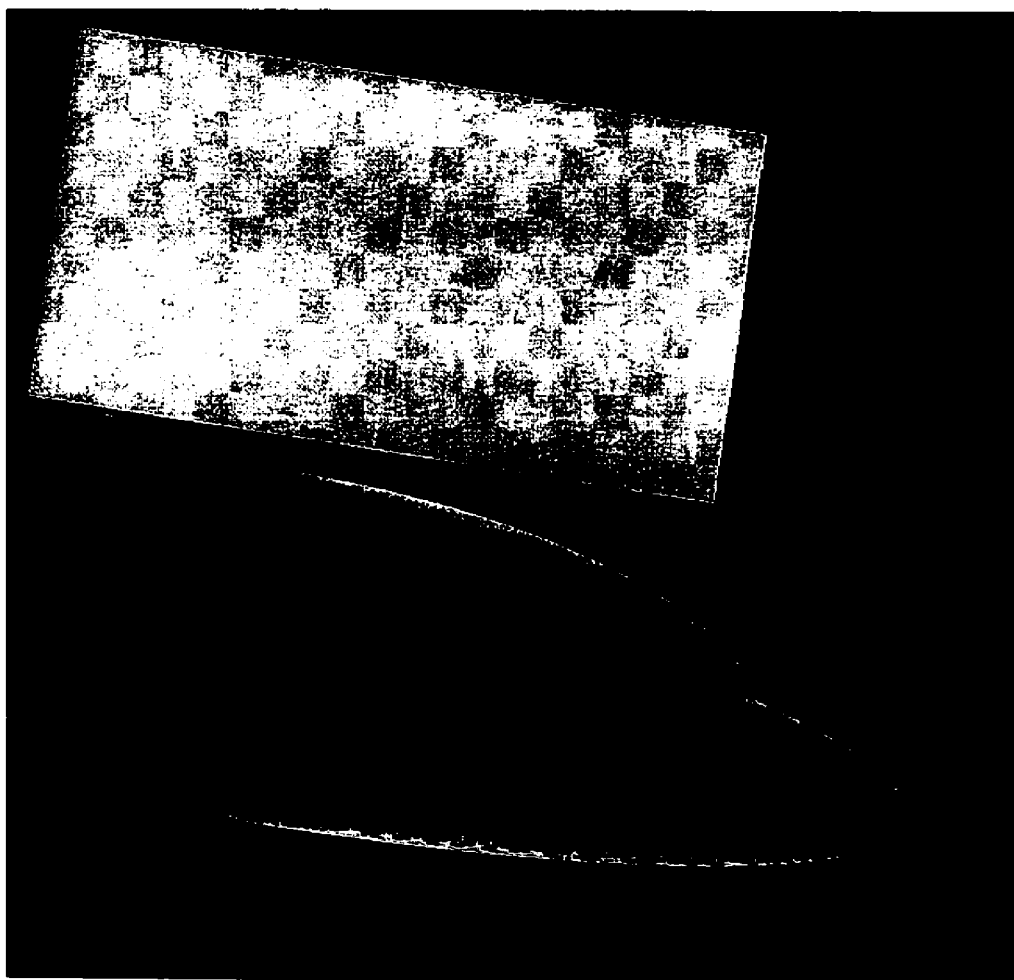


Figure 8.3: Elimination des quatres octants frontières et projection des noeuds des faces des octants frontières en face de l'objet.

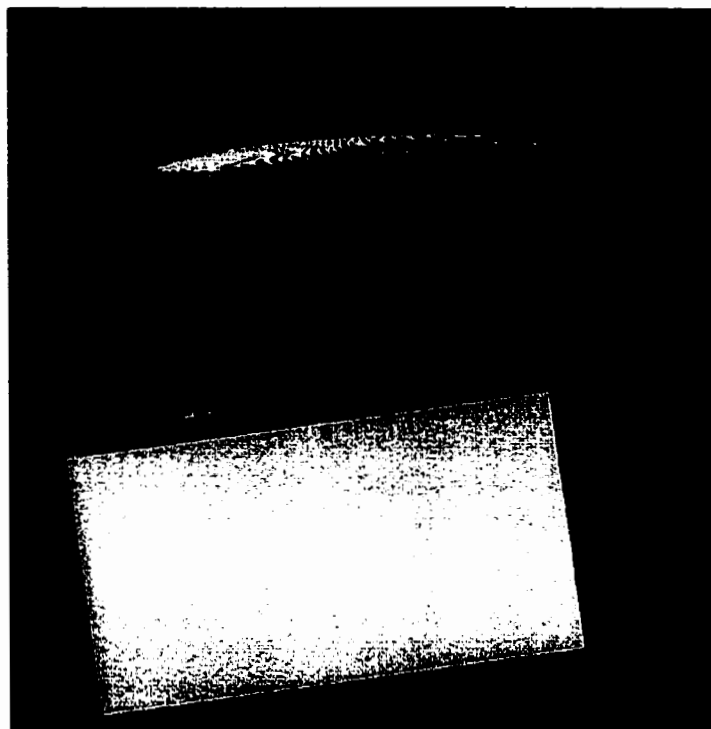


Figure 8.4: Rotation de la figure (8.3).

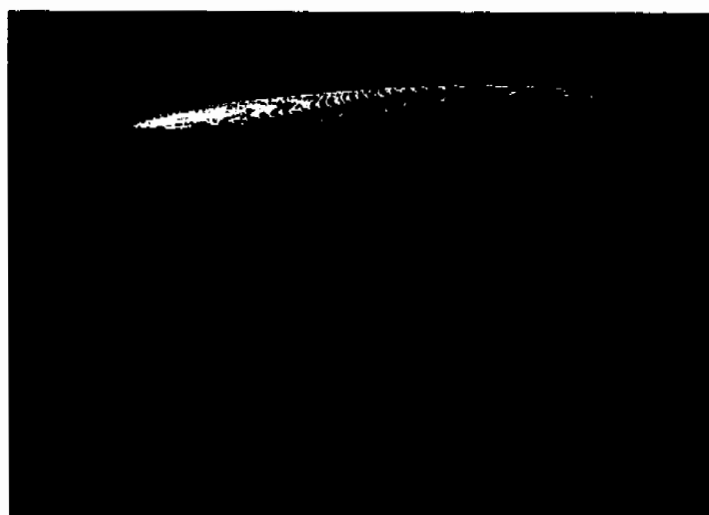


Figure 8.5: On isole les faces des octants frontières en face de l'objet pour mieux observer la précision de l'algorithme de projection.

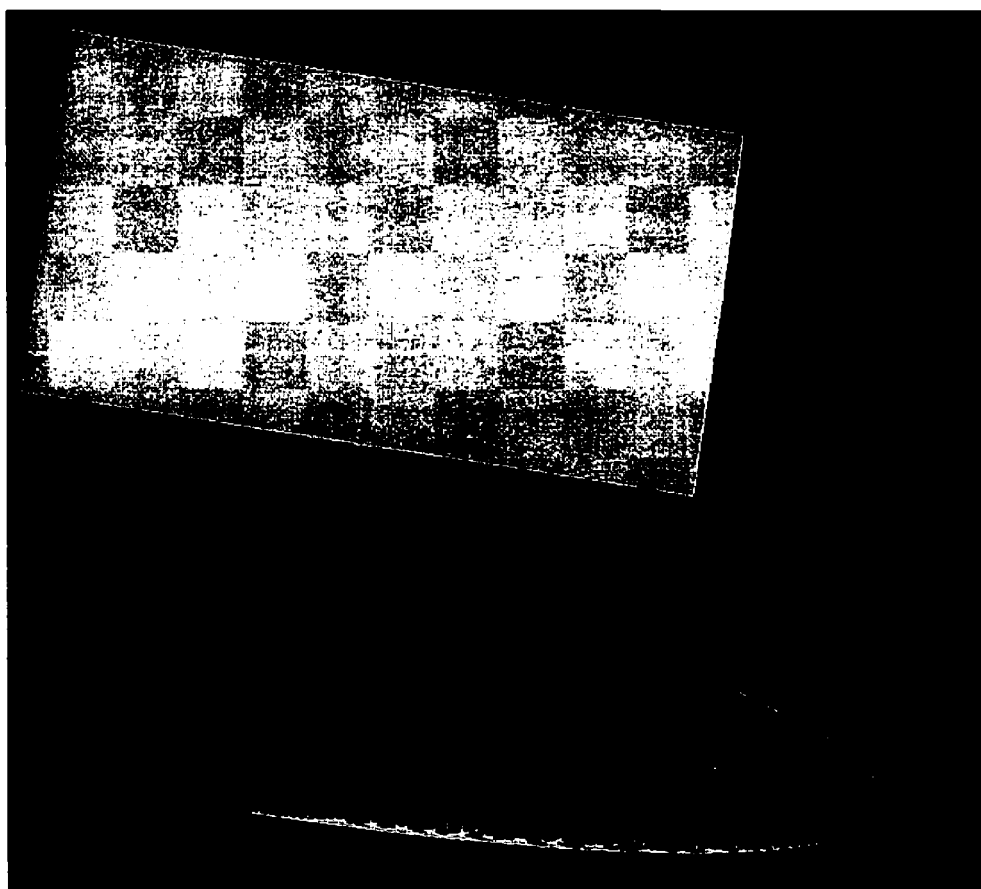


Figure 8.6: Génération de la couche interne de peau.

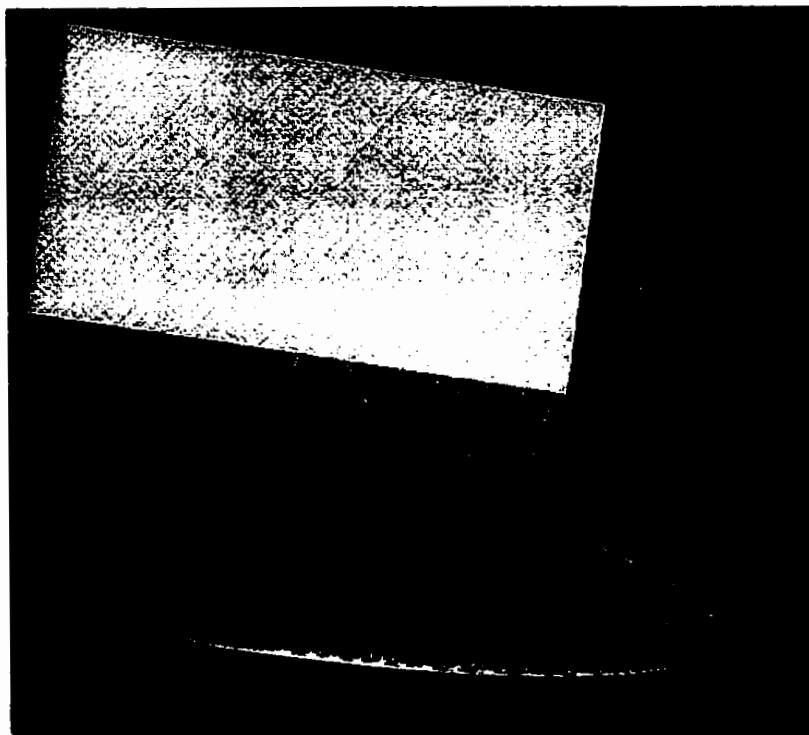


Figure 8.7: Génération de la deuxième couche de peau.

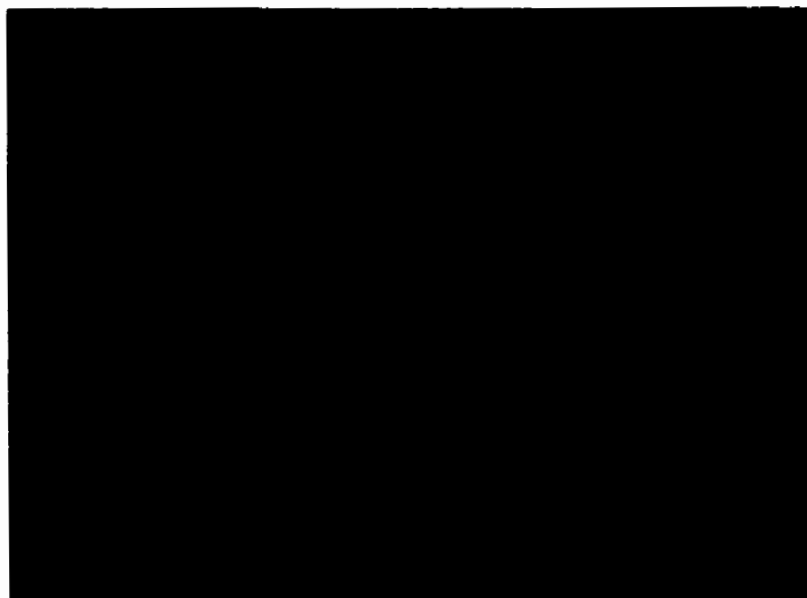


Figure 8.8: Visualisation des deux couches de peau. Dans cette figure nous avons enlevé le maillage extérieur.

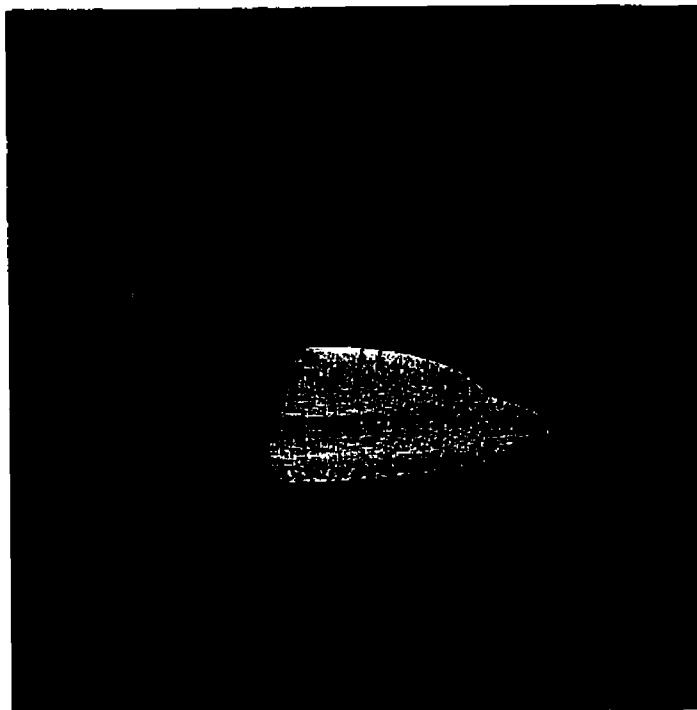


Figure 8.9: Visualisation de la projection des noeuds des octants frontières en face de l'objet pour une décomposition hiérarchique de niveau 2.



Figure 8.10: Isolation des faces des octants frontières en face de l'objet de la figure (8.9).

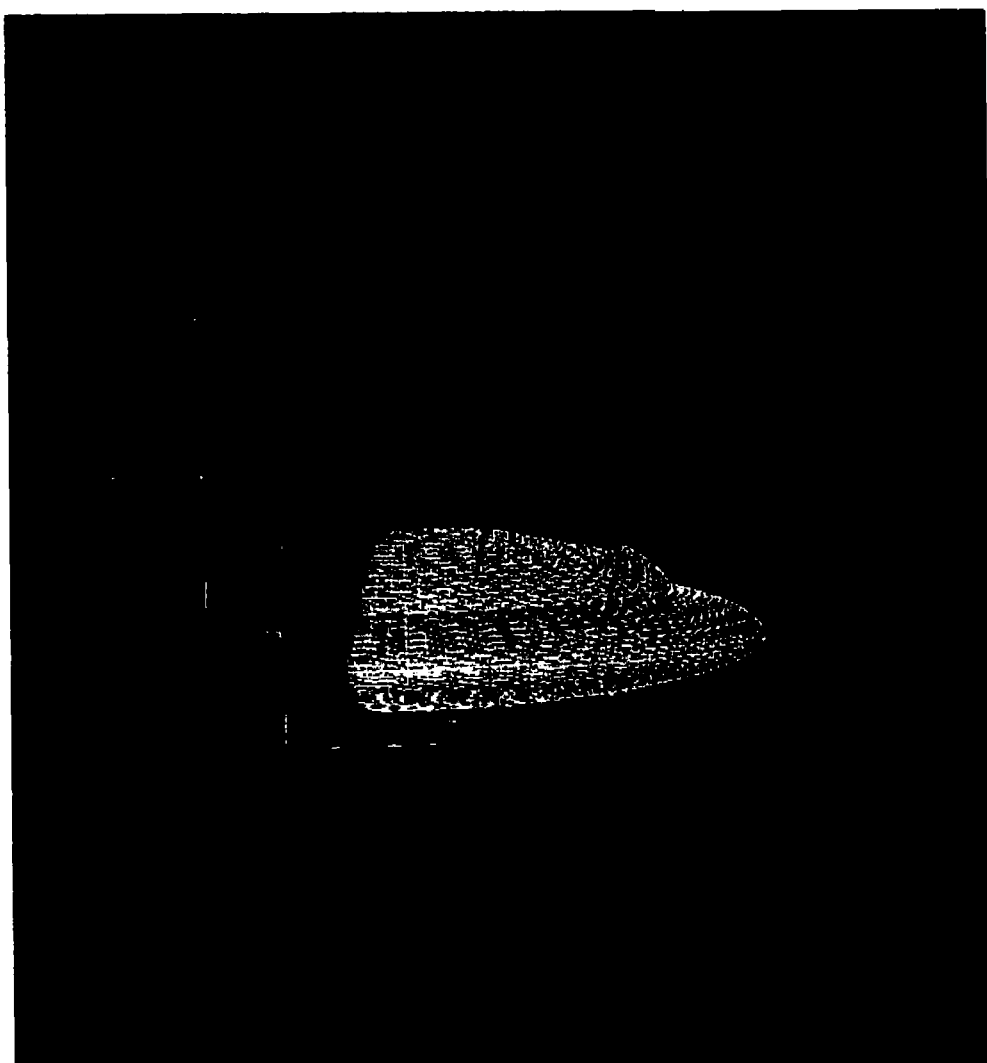


Figure 8.11: Figure illustrant la génération de la première couche de peau.

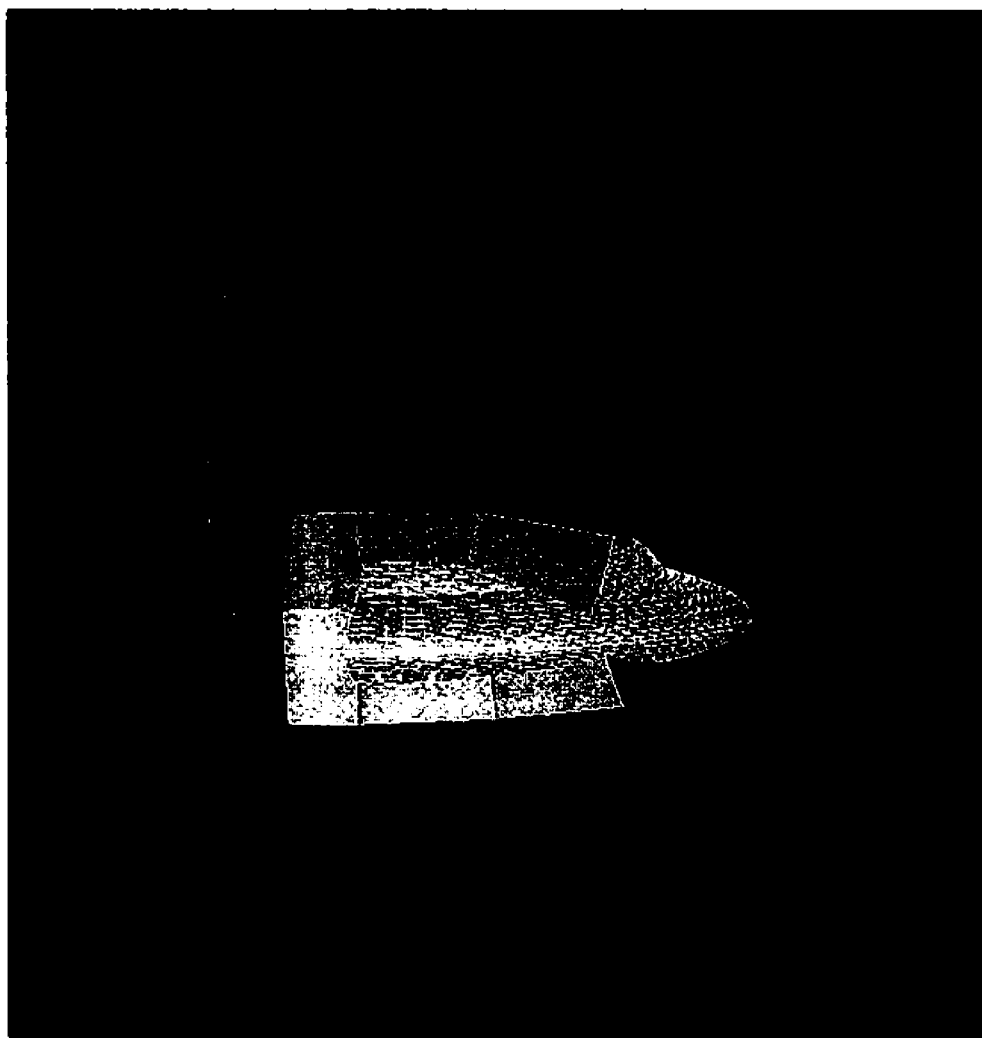


Figure 8.12: Une visualisation de côté de la structure d'ensemble comprenant le maillage de peau et le maillage extérieur.

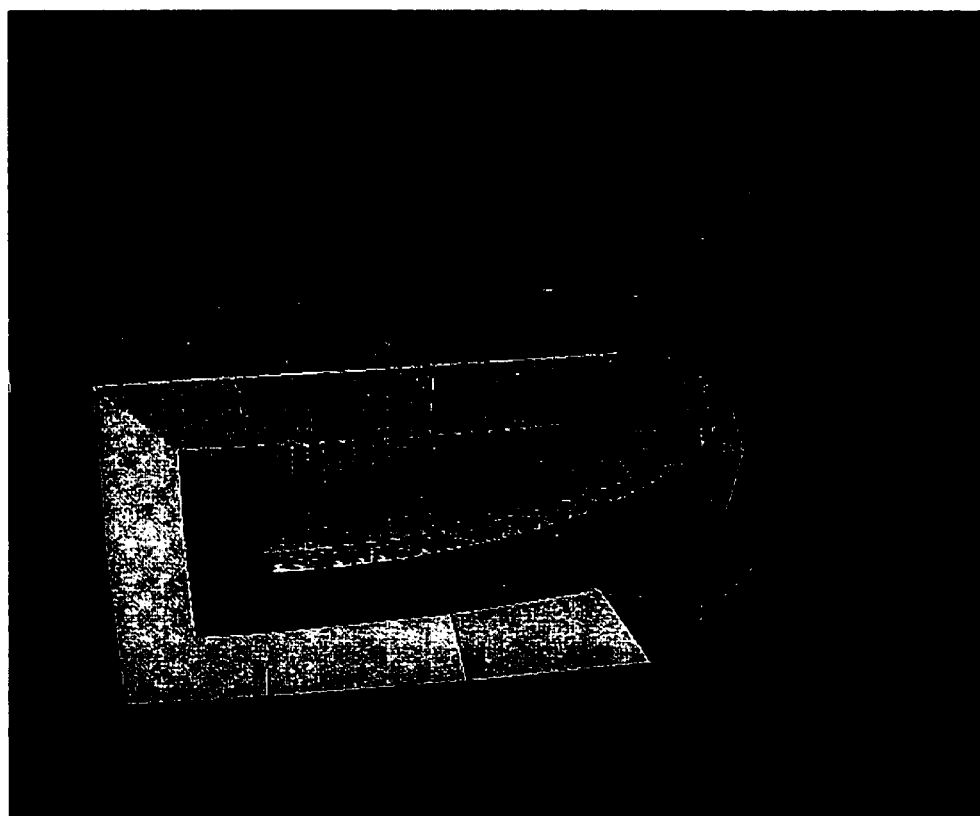


Figure 8.13: Visualisation de la couche intermédiaire dans le cas de la génération de 3 couches de peau.

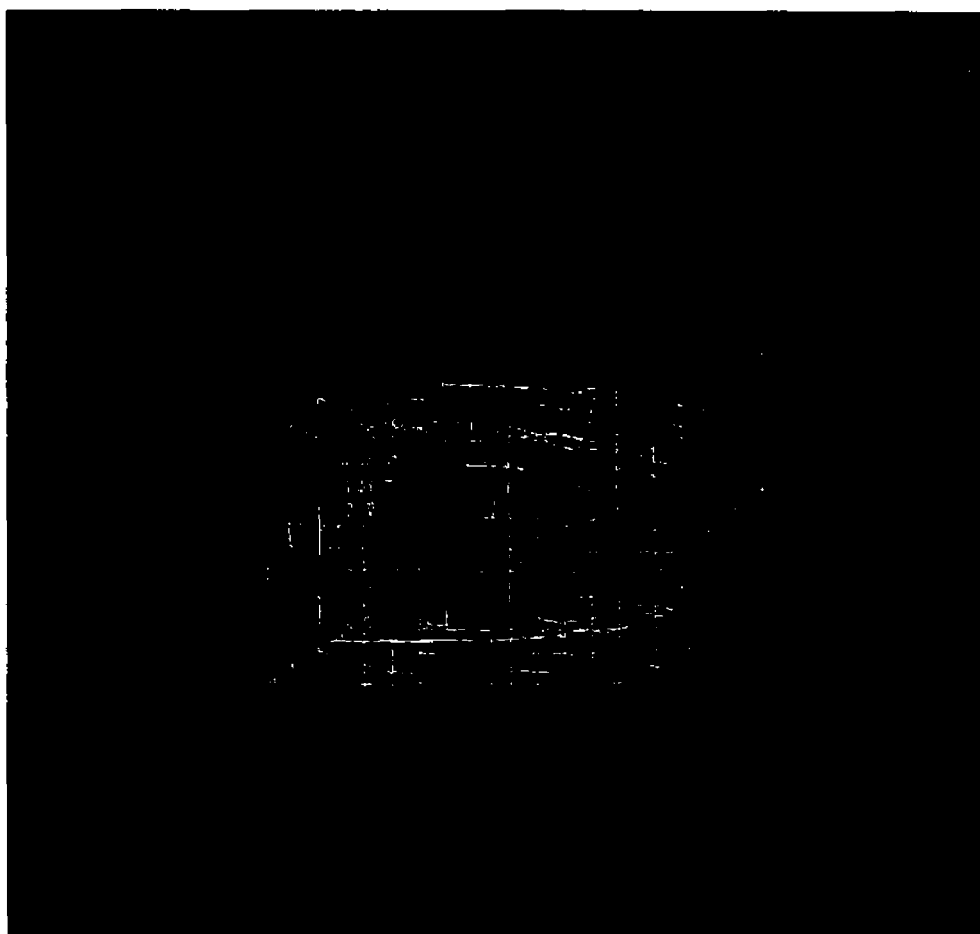


Figure 8.14: Illustration complète des 3 couches de peau et du maillage extérieur de la figure (8.13) en représentation fil de fer.

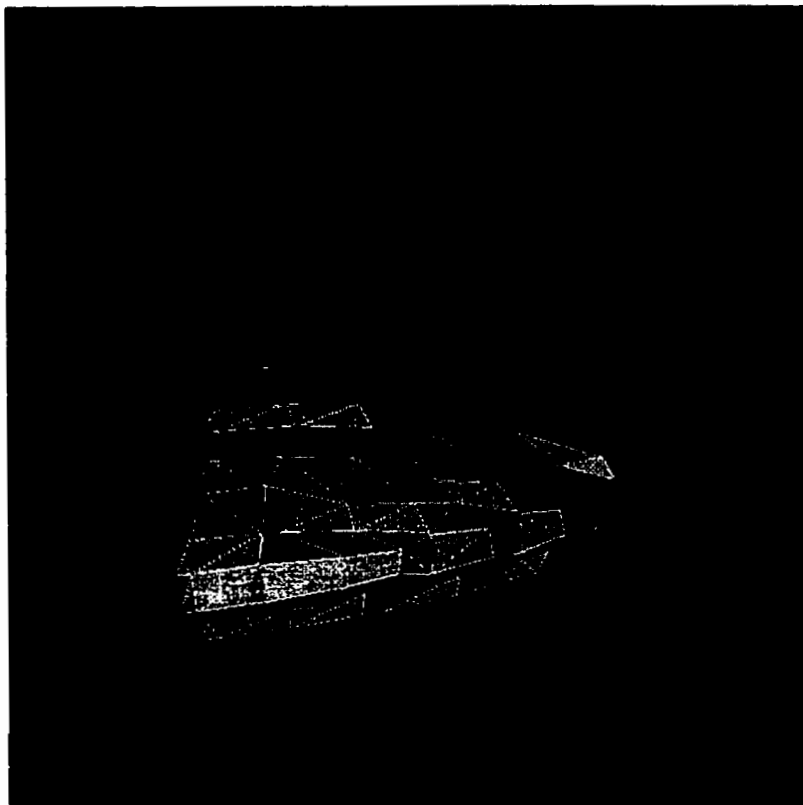


Figure 8.15: Une visualisation des barres de projection après la construction d'une première couche de peau.

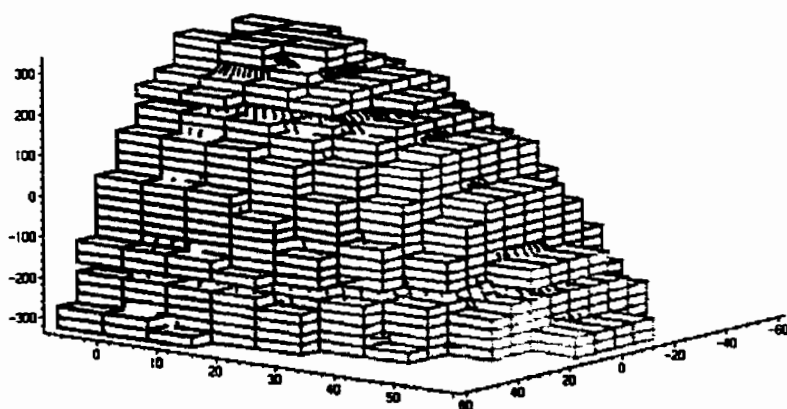


Figure 8.16: Décomposition octree générale de tout le nez de l'avion.

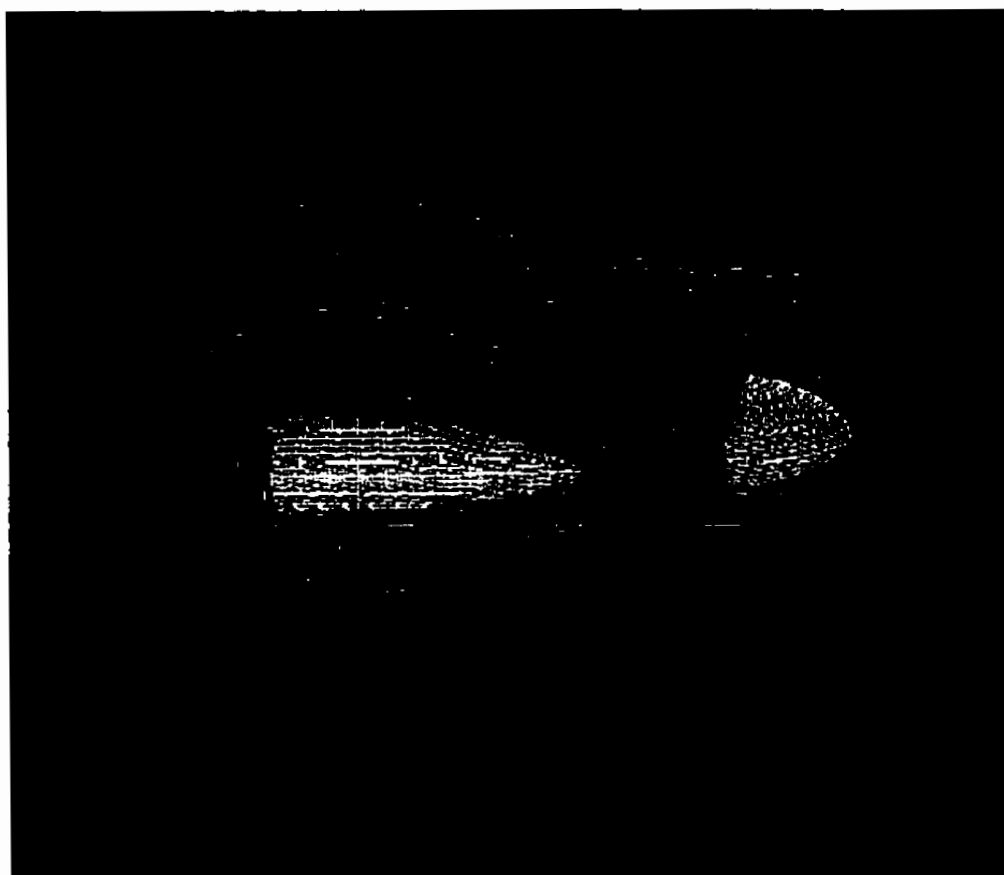
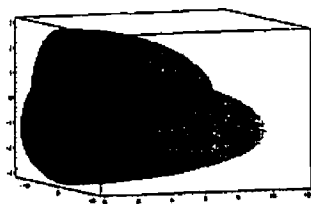
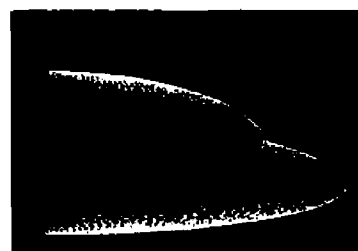


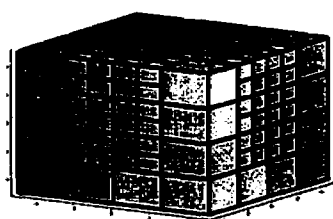
Figure 8.17: Zoom de quatre couches de peau générées autour du nez.



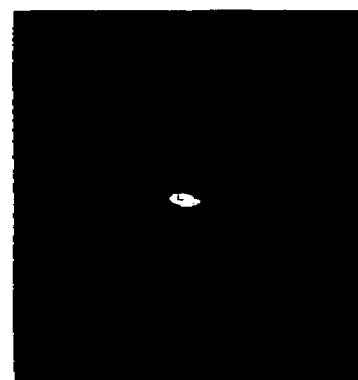
(a) Tête de l'avion modélisée par superellipse, en style patchcontours.



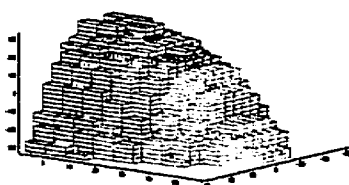
(b) Maillage triangulaire du nez de l'avion.



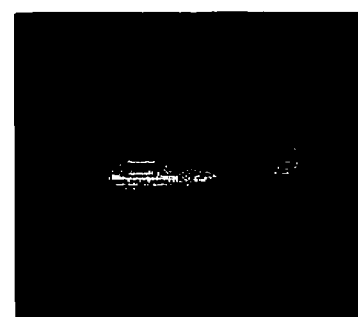
(c) Décomposition octree 'cachée' (par opposition à la décomposition transparente de fig. (d)) du nez de l'avion.



(d) Décomposition octree 'transparent' du nez de l'avion, que l'on peut observer au centre du cube univers.



(e) Représentation octree du nez de l'avion.

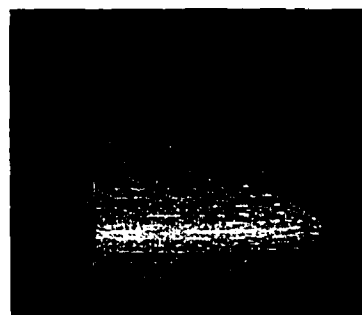


(f) Une visualisation de 4 couches de peau autour du nez de l'avion.

Figure 8.18: Visualisation de quelques étapes de la génération de maillage de peau autour du nez de l'avion.



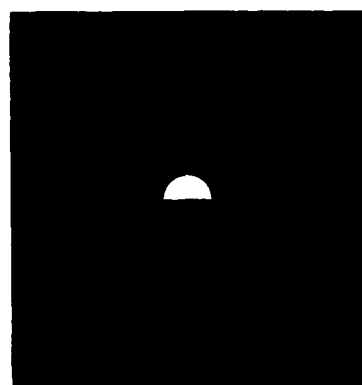
(a) Couche de peau autours du nez avion.



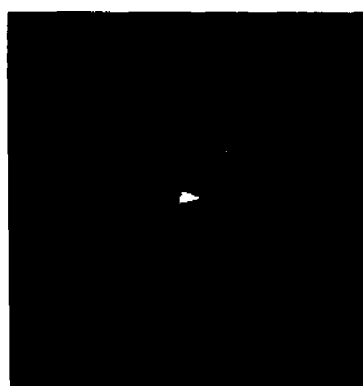
(b) Un zoom sur les couches successives de peau.



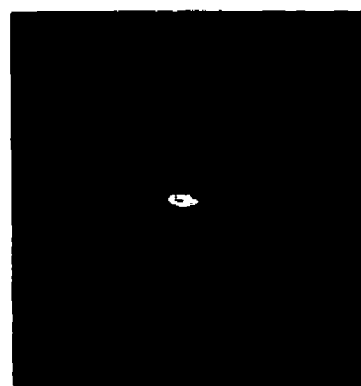
(c) Couches de peau et traces de projection



(d) Visualisation dans le plan xy du processus de décomposition.

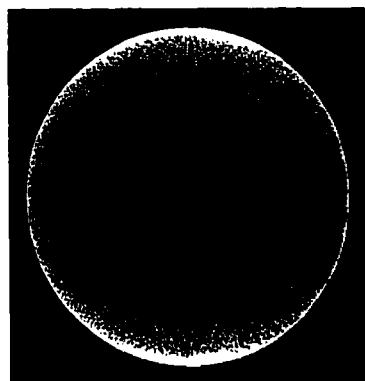


(e) Visualisation dans le plan yz de la décomposition octree du nez de l'avion.

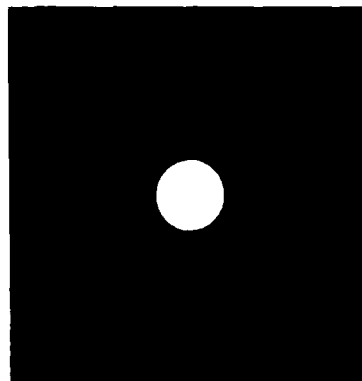


(f) Décomposition octree complète et 'transparente'.

Figure 8.19: Visualisation de couches de peau, suite.



(a) Représentation frontalière de la sphère.



(b) Vision Quadtree de la sphère.



(c) Représentation Octree de la sphère.



(d) Sphère et traces de projection.

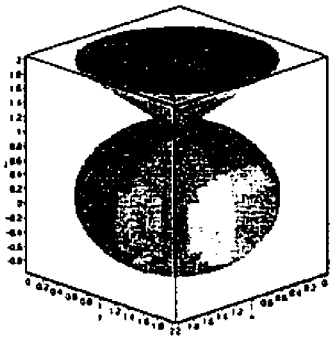


(e) Décomposition Octree "transparente" d'une sphère.

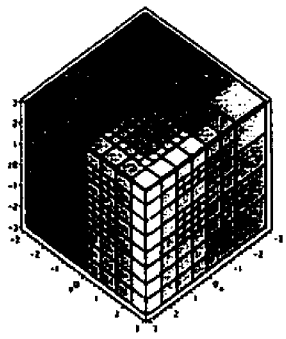


(f) Sphère représentée uniquement la trace de la projection des faces en face.

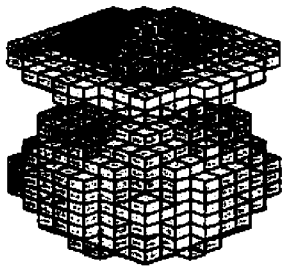
Figure 8.20: Simulation Octree pour une sphère.



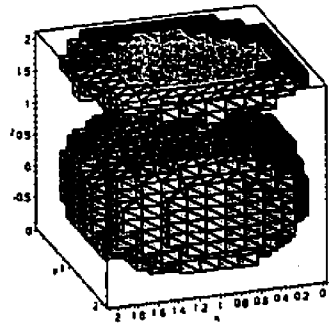
(a) Objet composé d'un cône-sphère et d'une sphère.



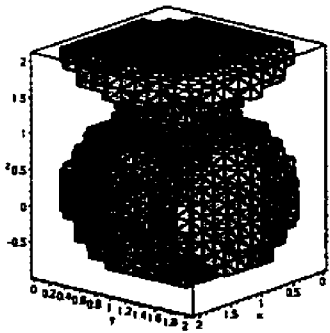
(b) Cône-sphère en décomposition Octree, avec respect du rapport 2:1.



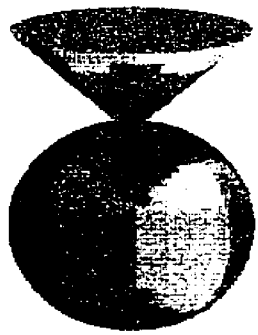
(c) Représentation Octree de la cône-sphère.



(d) Faces des octants frontière en face de l'objet. On remarque la présence de l'objet à l'intérieur.

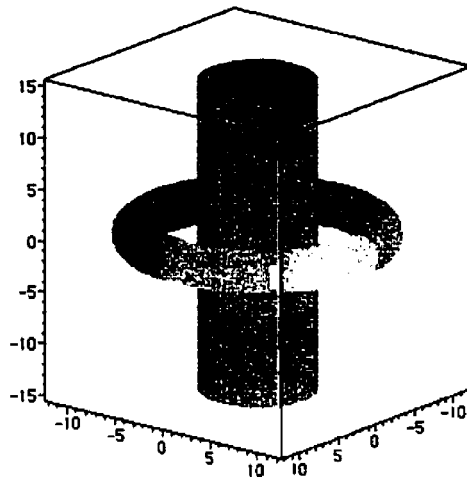


(e) Face des octants frontières en face. l'objet à été volontairement enlevé.

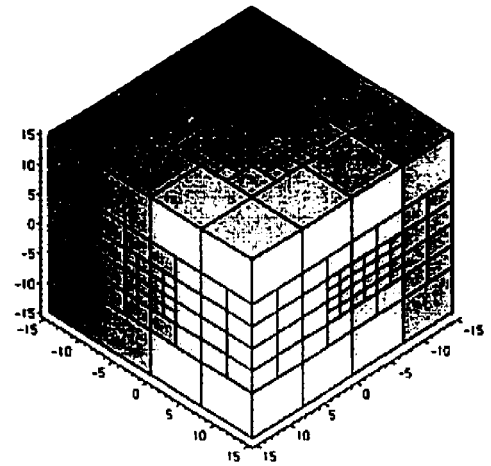


(f) cône-sphère sans référentiel.

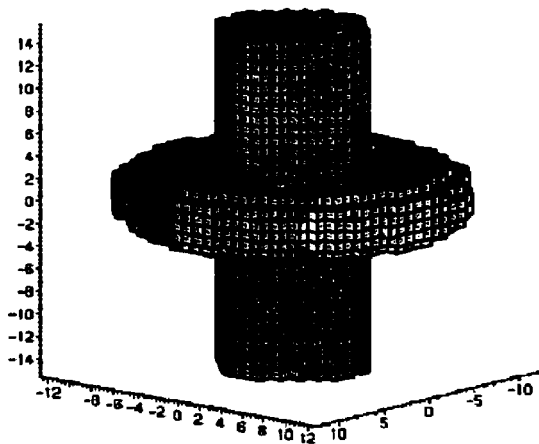
Figure 8.21: Simulation Octree d'un objet composé d'une sphère et d'un cône.



(a) Objet composé d'un tore et d'un cylindre.



(b) Représentation duale du tore-cylindre.



(c) Représentation Octree du tore-cylindre.

Figure 8.22: Simulation octree tore-cylindre.

Chapitre 9

Perspectives et travaux futurs

À travers l'expérience acquise tout au long de ce processus, nous pensons que le travail peut-être étendu et complété dans les directions suivantes et qui n'ont pas été abordées ici:

- Proposition d'un algorithme de projection pour géométries concaves.
- Proposition d'un critère d'arrêt de la décomposition qui *colle* à la physique.
- Proposition d'une accélération parallèle de la décomposition octree.

9.1 Proposition de récupération de coins concaves

L'algorithme de projection tel que proposé dans la section (7.2.3) fonctionne bien sur des géométries convexes, à tel point que l'on peut représenter les objets uniquement par la trace de la projection des noeuds des faces des octants frontières en face de l'objet tel que visualisable par exemple aux figures (7.11), (7.12). Pour traiter adéquatement les géométries concaves l'algorithme devra être modifié et tenir compte des coins concaves. Pour cela on devra y inclure deux opérations supplémentaires qui sont: une opération de test de détection des coins concaves, et ensuite une autre sur les arêtes frontière en face de l'objet. On peut utiliser le critère suivant: si l'angle entre les normales des deux triangles que partage cette

arête dépasse un certain seuil, cette arête devra être conservée pour *re-projection* sur la surface de l'objet. Par cette approche nous pensons que la topologie des noeuds frontières coïncidera avec la topologie de la surface de l'objet tel que illustré par la figure (9.1) ci-dessous.

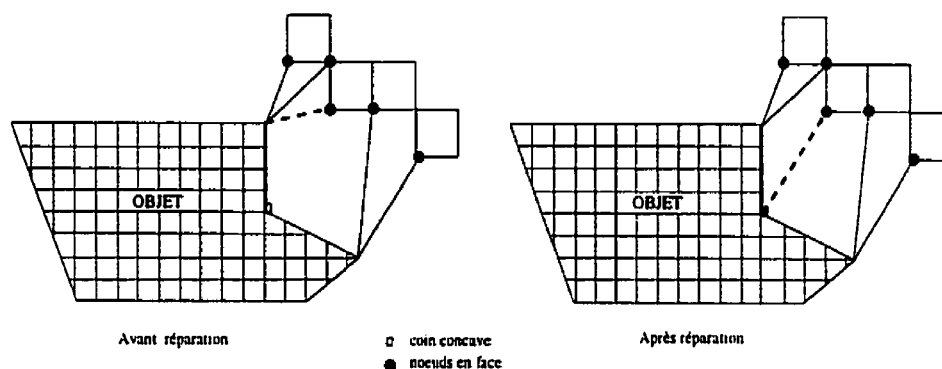


Figure 9.1: Illustration de la récupération d'un coin concave en face du front de projection.

9.2 Proposition d'une avenue pour un critère physique automatique d'arrêt

En partant d'un objet qui a été modélisé par superellipses et programmé et discrétisé en langage MapleV, l'algorithme de calcul d'intersection présenté à la section (7.2.1) était la suivante:

- liste 1 - Octants sur la frontière
- liste 2 - Octants qui ne sont pas sur la frontière
- Génération de l'octant de départ (hors de la frontière (liste 2))
- POUR k fois (k niveaux de subdivisions fixé au départ)
 - POUR chaque octant dans la liste 2
 - SI l'octant est sur la frontière ALORS
 - Transferer de la liste 2 à 1
 - SINON
 - Diviser en huit dans la liste 1

De cet algorithme, on peut remarquer que le critère d'arrêt de la décomposition Octree est fixé au départ et de manière arbitraire. La proposition de critère *d'arrêt physique et automatique* vise à jumeler la résolution de l'équation de Navier-Stokes à l'épaisseur de la couche limite dans le cadre de la simulation d'un problème CFD. Pour cela partons de l'équation de Navier-Stokes ci-dessous:

$$\text{div} \vec{u} = 0 \quad (9.1)$$

$$\frac{\partial \vec{u}}{\partial t} + (\nabla \vec{u}) \cdot \vec{u} = -\frac{1}{\rho} \nabla p + \nu \Delta \vec{u} + \vec{f} \quad (9.2)$$

où les inconnus sont les champs de vitesse \vec{u} et de pression p , la masse volumique ρ , les forces extérieures \vec{f} et la viscosité ν sont données. Considerons le cas 2D pour un fluide incompressible et dans le système de coordonnées cartésien, elles peuvent s'exprimer sous la forme:

$$u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = -\frac{1}{\rho} \frac{\partial p}{\partial x} + \nu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \quad (9.3)$$

$$u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} = -\frac{1}{\rho} \frac{\partial p}{\partial y} + \nu \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) \quad (9.4)$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0. \quad (9.5)$$

L'hypothèse de la couche limite conduit à la comparaison de l'ordre de grandeur des termes d'inertie et des termes de cisaillement. Si on compare l'ordre du terme $u \frac{\partial u}{\partial x} \approx v \frac{v}{L}$ avec celui de $\nu \left(\frac{\partial^2 u}{\partial y^2} \right) \approx \nu \frac{1}{\delta} \frac{v}{\delta}$, on obtient:

$$\frac{v^2}{L} \approx \frac{\nu v}{\delta^2} \quad (9.6)$$

$$\left(\frac{\delta}{L} \right)^2 \approx \frac{\nu}{vL} \quad (9.7)$$

$$\frac{\delta}{L} \approx \frac{1}{R\sqrt{e}} \quad (9.8)$$

Par hypothèse, la couche limite satisfait $\frac{\delta}{x} \ll 1$. Alors, les équations sont valables lorsque R_e est grand. L'approche que nous préconisons consiste à laisser la décomposition se faire récursivement à partir de la racine comme on peut le visualiser à partir des figures (6.12) et (4.4), sans imposer de critère d'arrêt arbitraire tant et aussi longtemps qu'un critère du type de l'équation (9.6) qui lie l'épaisseur

de la couche limite, le nombre de Reynolds et la longueur d'échelle caractéristique n'aurait pas été satisfait. Une fois ce critère d'arrêt lié à la physique atteint, on peut alors enlever des octants qui sont directement en contact avec l'objet, créant alors un vide que l'on peut observer sur la figure (7.6). On peut par la suite passer aux étapes de projection et de génération de couches structurées de peau. Avec ce critère d'arrêt, on aurait pour un maillage qui *colle* à la physique, un critère d'arrêt automatique de la décomposition qui lui même colle à la nature physique du problème et non plus fixé de manière arbitraire.

9.3 Proposition pour une accélération de la décomposition Octree

Par sa nature, la méthode des octree est une technique qui peut devenir très lourd: des tests préliminaires sur MapleV nous ont clairement indiqué que la constructions d'un octree par des méthodes inappropriées peut facilement prendre des heures, d'où la construction la classe Array de la section (7.3). Nous pouvons aller encore plus loin pour accélérer les calculs, et dans cette section nous proposons une avenue pour cette accélération de la décomposition et elle consiste à faire un jumelage entre la nature octree du processus de décomposition et sa parallélisation. C'est une avenue qui mérite d'être explorée plus en profondeur surtout dans le contexte actuel marqué par des progrès importants en terme de rapidité, de puissance et de baisse continue du prix de la mémoire et des autres composants informatique. Utiliser les ordinateurs parallèles pose la question de l'endroit stratégique de l'utilisation de ces machines et aussi de quel type de machine à utiliser. Pour répondre à ces deux questions, partons avec un processus CFD typique en examinant la figure (9.2) et aussi en retournant à l'idée de base de la méthode des Octree par un examen des figures (6.12) et (4.4). On peut remarquer qu'il est théoriquement possible de bâtir chaque sous-arbre d'octants indépendamment du reste une fois la première étape réalisée. Cette propriété de la décomposition géométrique peut être exploitée pour construire une méthode de génération de maillage hybride en parallèle. Pour répondre à la deuxième question nous pensons que l'ordinateur de

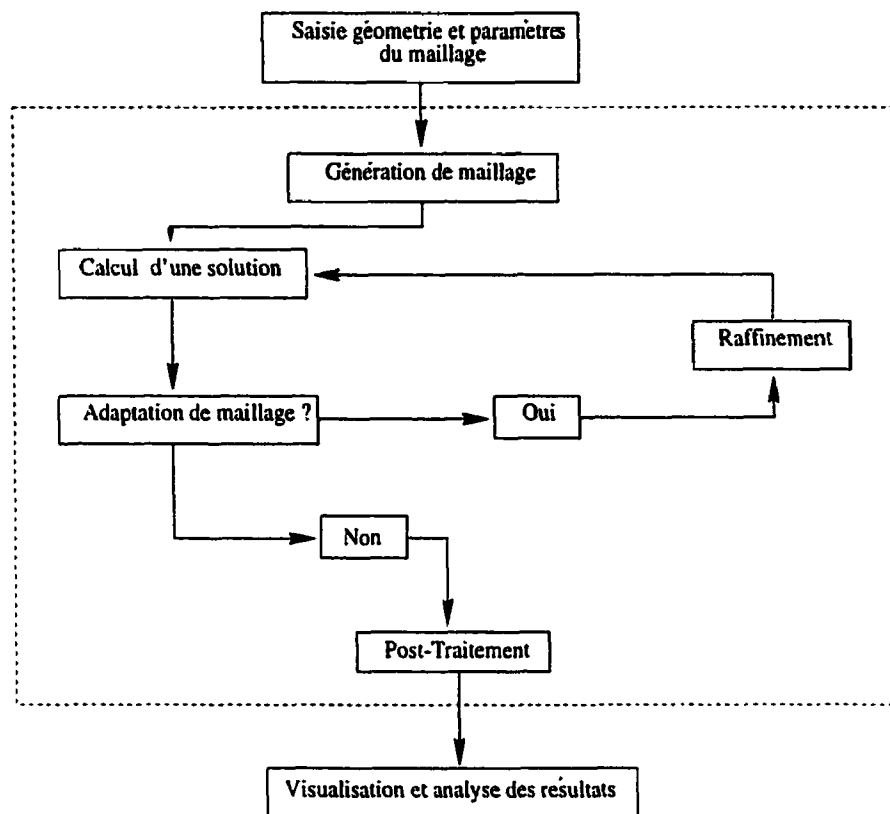


Figure 9.2: Illustration d'un processus CFD typique.

type M.I.M.D. (Multiple Instructions stream Multiple Data stream) peut être un candidat intéressant.

9.3.0.1 Méthodologie Proposée

Pour l'utilisation concrète de la machine M.I.M.D. Nous proposons une première étape de décomposition classique de l'octant racine par un processeur maître (traitement séquentiel), et à la deuxième étape et étapes subséquentes chaque octant pourrait continuer à être décomposé par des processeurs indépendants (processeurs esclaves, en traitement parallèle) comme illustré par la figure (9.3). La démarche proposée peut-être encore améliorée en y rajoutant un aspect adaptatif et en exploitant toujours l'indépendance des octants. Pour chaque octant on connaît l'octant père donc son niveau de subdivision, On peut donc raffiner localement en subdivisant l'octant père en huit octants ou bien on peut monter vers la racine de l'arbre pour

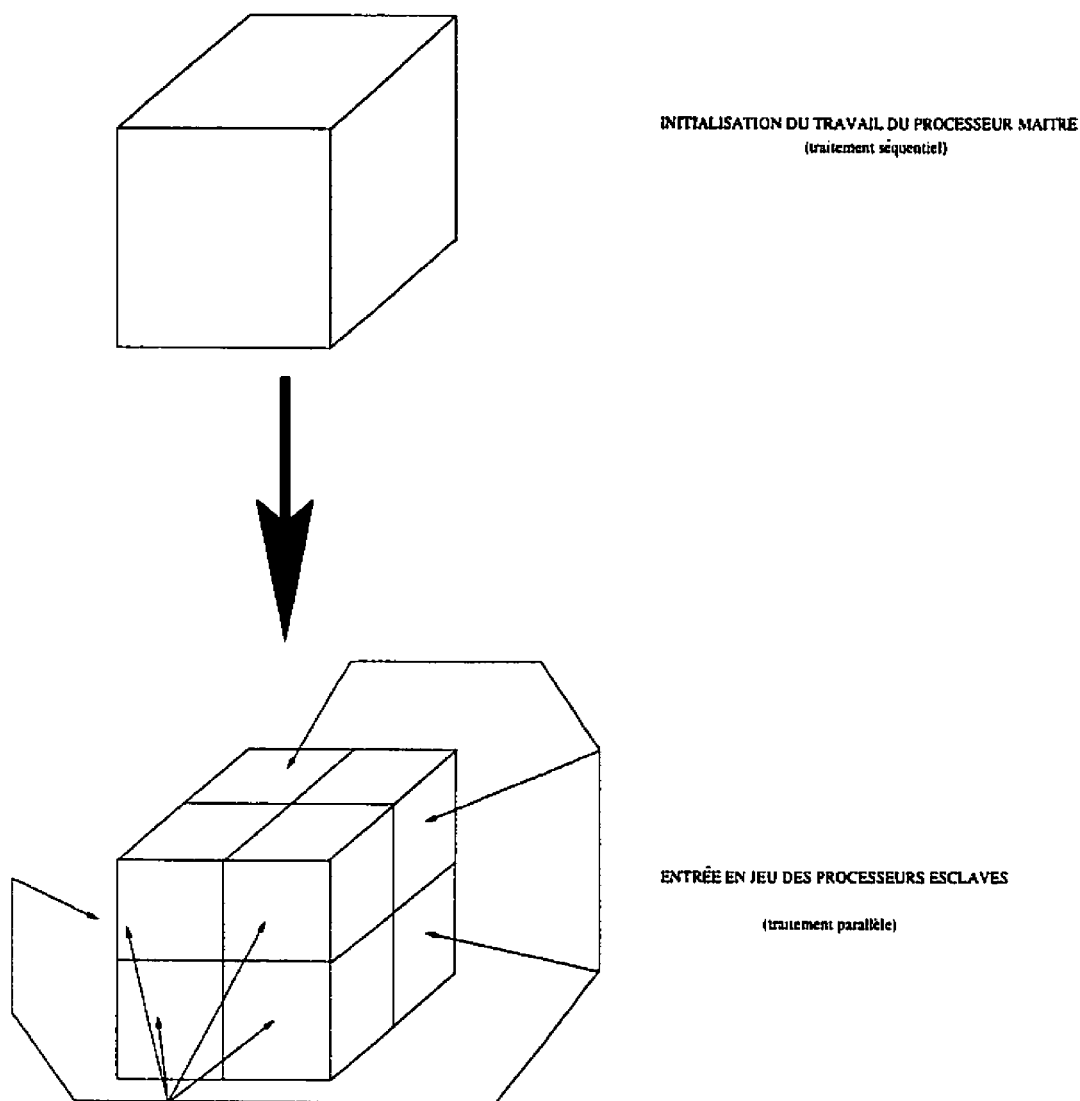


Figure 9.3: Illustration de la décomposition parallèle de chaque octant par un processeur indépendant.

avoir le maillage le plus gros c'est-à-dire faire un deraffinement sans toucher aux autres zones du maillage. Cette remarque peut être couplée avec un estimateur d'erreur pour trouver les zones critiques et alors raffiner ou deraffiner localement. De cette façon on peut réduire le coût du calcul car on n'a pas besoin de remailler. On peut choisir comme critère d'adaptativité un estimateur d'erreur basé sur la courbure des objets (comme défini à la section (B.7.3)). Nous proposons l'algorithme

suiuant ci-dessous qui peut s'insérer dans un cadre global de simulation CDF comme illustré par la figure (9.2).

Saisie objet première étape de décomposition.

Si l'objet est présent

et la courbure est au dessus d'un certain seuil

et critère d'arrêt physique non atteint

faire une allocation de processeur à ce octant

décomposer cet octant en parallèle.

Sinon

Si objet est non présent continuer recherche

marquer cet octant

Fin Si

Conclusion

La signification de notre travail doit se placer dans un cadre *exploratoire*. Elle porte sur la génération de maillages de peau par une technique de décomposition spatiale hiérarchique et se situe dans la continuité de la nouvelle tendance de la génération des maillages qui veut que ceux-ci colle à la physique du problème. Dans cette thèse, notre contribution principale aura été de démontrer pour la première fois la faisabilité du passage en 3D de l'approche de Smith-Leschziner. Pour l'extension en 3D, des problèmes informatiquement difficiles de conception et d'implémentation de structures de données internes, d'algorithmiques et d'ordonnancement ont été résolus. Les structures de données et algorithmes développées ont fait l'objet d'une mise en œuvre informatique ayant permis de produire des maillages de peau autour de quelques formes géométriques.

La méthode de la décomposition spatiale est efficient pour le traitement des géométries complexes de manière globale, et peut être une technique intéressante pour automatiser entièrement le processus de génération de maillages. On remarquera que la plupart des exemples de géométries que nous avons présenté sont de formes convexes. Ceci s'explique par le fait qu'en développant cette nouvelle méthodologie, nous avons commencé par ces formes relativement simples avec une évolution future possible vers des formes fortement concaves ou des espaces encombrés, des géométries complexes de type industriel, etc... des aspects qui n'ont pas été abordés. Pour le cas spécifique de la récupération des coins concaves nous avons proposé une technique qui permettra d'ajuster, de corriger les octants frontières aux coins de l'objet.

On peut aussi s'interroger sur la validité du rapport 2:1 c'est-à-dire de la

contrainte d'un seul niveau de différence sur les niveaux minimaux de différence de maillage entre octants voisins dans le contexte de la génération de maillages hybrides. La quasi totalité des références que nous avons mentionné ont été développés dans un contexte d'éléments finis et les auteurs de ces articles appliquent *naturellement* le principe car elle permet une transition douce et lisse entre éléments de niveaux de subdivision différents. Dans le cadre de notre travail, l'application du principe reste en option. On peut remarquer qu'elle n'a aucune influence sur la décomposition (la représentation) octree de l'objet ainsi que sur les autres aspects critiques comme la projection, l'ordonnancement etc... Si on décide de l'appliquer, l'étape de l'imposition de la différence d'un seul niveau, normalement doit être faite après la génération de l'arbre octree. Pour améliorer le temps d'exécution du programme et éviter l'augmentation des voisinages des octants, on vérifie cette condition à chaque fois qu'on subdivise un octant dans le processus de génération de l'arbre octree.

Dans un processus CFD typique l'acquisition, la saisie de la géométrie peut prendre énormément de temps. Cette difficulté a été contournée dans le cadre de notre étude par l'intégration du langage de programmation symbolique MapleV qui a permis d'accélérer la phase de modélisation géométrique et de discrétisation des objets sans être trop exigeant sur la quantité de mémoire nécessaire. MapleV est un système ouvert qui contient un langage de programmation fonctionnel performant et flexible, ce qui nous a permis de créer de nouvelles fonctionnalités selon notre besoin et de les incorporer ensuite dans le programme. Le seul problème est qu'il est un langage de programmation interprété ce qui provoque une grande lenteur dans l'acquisition de la structure discrète de l'objet lors de l'application des routines *grid* et *convert*. L'arrivée très récente sur le marché de la nouvelle version (Maple6.1) dotée d'un compilateur et permettant de faire du calcul numérique au même titre que MathLab par exemple permettra à notre avis de résoudre partiellement ce problème.

La génération de maillages est un aspect important et laborieux de la CFD et autres domaines de recherche. Notre contribution principale a été d'étendre la formulation existante en 2D de Smith et de Leschziner en une formulation 3D. À la différence de Smith et de Leschziner nous n'avons exécuté aucun calcul d'écoulements. Nous croyons que cet arrangement pourrait être très utile lorsque

jumellé à un solveur approprié de zones d'écoulements et particulièrement si le solveur devait employer un arrangement établi sur les structures que nous avons déjà développées. Le problème étendu de Smith et de Leschziner serait réellement résolu en 3D et on aura alors la preuve directe de la supériorité ou non de cette approche par rapport aux méthodes existantes.

Références

- AFTOSMIS, M. (1995). Emerging CFD Technologies and Aerospace Vehicle Design. Rapport technique, NASA, Lewis Research Center.
- BABUSKA, I. et AZIZ, A. (1976). On the angle condition in the finite element method. *SIAM.J.Numer.Anal.*, 13.
- BAKER, T. (1985). Automatic mesh generation for complex three-dimensional regions using a constrained delaunay triangulation. *Engineering With Computers*.
- BARR, A. (1981). Superquadrics and angle-preserving transformations. *Ieee Computer Graphics and Applications*, 11-23.
- BERN, M. et EPPSTEIN, D. (1992). *Computing in euclidean geometry*.
- BERN, M., EPPSTEIN, D. et GILBERT, J. (1990). Provably good mesh generation. 105, 125-141.
- CASTILLO, J. (1991). *Mathematical aspects of numerical grid generations*, vol. 8. FAM.
- CHARLTON, E. et POWEL, K. (1997). An Octree Solution to Conservation-laws over Arbitrary Regions (oscar). *AIAA - 35th Aerospace Sciences Meeting & Exhibit*. Reno, NV, no. Paper AIAA-97-0198.
- CHARLTON, E. F. (1997). *An Octree Solution to Conservation-laws over Arbitrary Regions with Applications to aircraft Aerodynamics*. Thèse de doctorat, University of Michigan.

- CHEN, A. et KALLINDERIS, J. (1996). Adaptive hybrid (prismatic / tetrahedral) grids for 3-d incompressible flows. *AIAA - 34th Aerospace Sciences Meeting & Exhibit*.
- CHEW, P. (1989). Guaranteed-quality triangular meshes. Rapport technique TR-89-983, Cornell University.
- CIARLET, P. et RAVIART, P. (1972). General lagrange and hermite interpolation in \mathbf{R}^n with applications to finite elements methods. *Arch Rational Mech Anal*, 46.
- COIRIER, W. (1994). *An Adaptively-Refined, Cartesian Grid Algorithm for Solution of the Euler Equations*. Thèse de doctorat, The University of Michigan.
- COUPEZ, T. (1991). *Grandes Transformations et Remaillage Automatique*. Thèse de doctorat, École Nationale Supérieures des Mines de Paris.
- GEORGE, P. et BOROUCAKI, H. (1997). *Triangulation Delaunay et Maillage, application aux éléments finis*. Editions HERMES, Paris.
- GOLGOLAB, A. (1989). Maillages 3d automatiques pour géométries complexes. Rapport technique INRIA-1004, INRIA.
- GUIBAUT, F. (1998). *Un Maillage Structuré/Non-Structuré en Trois Dimensions*. Thèse de doctorat, Ecole Polytechnique De Montréal.
- JOE, B. (1995). Construction of three dimensional improved-quality triangulations using local transformations. *SIAM J. Scientific Comp.*, 1292-1307.
- KALLINDERIS, J., KHAWAJA, A. et MORRIS, H. M. (1995). Hybrid prismatic / tetrahedral grid generation for complex geometries. *AIAA95*. Reno, NV, no. Paper AIAA-95-0211.
- KALLINDERIS, J. et WARD, S. (1993). Prismatic grid generation for three dimensional complex geometries. *AIAA Journal*, Volume 31.
- KHAWAJA, A., MORRIS, H. M. et KALLINDERIS, J. (1995). Hybrid grid for viscous flows around complex 3-d geometries including multiple bodies, aiaa paper 95-1685, cp san francisco. 25.

- MARCUM, D. (1995). Generation of unstructured grids for viscous flow applications, 129–151.
- MINYARD, T. et KALLINDERIS, Y. (1998). Octree partitioning of hybrid grid for parallel adaptive viscous flow simulations. *International Journal for Numerical Methods in Fluids*, 26, 57–78.
- MITCHELL, S. et VAVASIS, S. (1992). Quality mesh generation in three dimension. *Proceedings of the Eight Annual Symposium on Computational Geometry*.
- MORRIS, H. M. et KALLINDERIS, J. (1996). A combined octree advancing front method for tetraedra and anisotropic surface meshes. *Computer Methods in Applied Mechanics and Engineering*, 96, 117–129.
- MOULARD, L. (1994). *Optimisation de Maillages Non-Structurés: Application à la génération, à la correction et à l'adaptation*. Thèse de doctorat, Université Joseph Fourier-Grenoble1.
- NAKASHI, K. (1993). External viscous flow computations using prismatic grids. S. Verlag, éditeur, *Lecture Notes in Physics 414, Thirteenth Int. Conf. on Num. Meth. In Fluid. Dynam.* 280–284.
- PARTHASARATHY, V., KALLINDERIS, Y. et NAKAJIMA, K. (1995). Hybrid adaptation method and directional viscous multigrid with prismatic tetrahedral meshes. *Computer Methods in Applied Mechanics and Engineering*, 32, 199–259.
- PERAIRE, J., MORGAN, K. et PEIRO, J. (1986). *Unstructured Finite Mesh Generation and Adaptive Procedures for CFD, in applications of Mesh generation to complex 3D configurations*, vol. VI. Prentice-Hall.
- PERAIRE, J., MORGAN, K. et PEIRO, J. (1990). Unstructured finite element mesh generation and adaptive procedures for CFD, in Application of Mesh generation to complex 3d configurations. *AGARD conference Proceedings*. No. 464.
- PIEGL, L. et TILLER, W. (1997). *The NURBS book*. Springer-Verlag.
- PIRZADEH, S. (1996). Progress toward a user oriented unstructured viscous grid generator. *AIAA - 34th Aerospace Sciences Meeting & Exhibit*, 15.

- PRATT, M. et GEISOW, A. (1986). Surface/surface intersection problems. J. Gregory, éditeur, *The proceedings of a conference of mathematics and its applications*. University of Manchester, 117–125.
- RHEINBOLT, W. et MESZTENYI, C. (1980). On a data structure for adaptive finite element mesh refinements. *Acm Transaction on Mathematical Software*, 6, 166–187.
- ROGERS, S. E., ROTH, R., NASH, S. M., BAKER, D., SLONICK, J. P., WHITLOCK, M. et CAO, H. V. (2000). Advances in overset CFD processes applied to subsonic high-lift aircraft 18th AIAA Applied Aerodynamics Conference 2000-4216, Denver, Colorado.
- SAMET, H. (1990a). *Applications of spatial data structures*. Addison Westley.
- SAMET, H. (1990b). *The design and analysis of spatial data structures*. Addison Westley.
- SCHNEIDER, R. (1996). A grid based algorithm for the generation of hexahedral element meshes. *Engineering With Computers*, 12, 168–177.
- SCHNEIDER, R., OBERSCHELP, W., KOPP, R. et BECKER, M. (1992). New and effective remeshing scheme for the simulation of metal forming processes. *Engineering With Computers*, 8, 163–176.
- SCHROEDER, W. et SHEPHARD, M. (1991). A combined octree / delaunay method for fully automatic 3-d mesh generation. *International Journal for Numerical Methods in Engineering*, 29, 37,55.
- SEDGEWICK, R. (1992). *Algorithms in C++*. Addison-Wesley.
- SHARIFI, H. (1996). *Maillage Automatique par la Méthode de quadtree/octree Modifiée*. Thèse de doctorat, Université Laval.
- SHEPHARD, M. et GEORGE, M. (1991). Automatic three dimensional mesh generation by the finite octree technique. *International Journal for Numerical Methods in Engineering*, 32, 709–749.

- SMITH, R. et JOHNSTON, L. (1995). A novel approach to engineering computations for complex aerodynamic flows, paper 95-0214. *AIAA - 33th Aerospace Sciences Meeting & Exhibit*, 271-285.
- SMITH, R. et LESCHZEINER, M. A. (1996). Automatic grid generation for complex geometries. *Aeronautical Journal*, 7-14.
- SMITH, R. et LESCHZINER, M. (1996). A novel cartesian grid method for complex aerodynamic cfd applications. *Proceedings of the fifth International Conference on Numerical Grid Generation in Computational Field Simulations*. 709-716.
- SOLINA, F. et BAJCSY, R. (1990). Recovery of parametric models from range images: The case for superquadrics with global deformations. *Ieee Transactions on Pattern Analysis and Machine Intelligence*, 12, 131,147.
- TCHON, K., HIRCH, C. et SCHNEIDER, R. (1997). Octree-Based Hexahedral Mesh Generation For Viscous Flow Simulations. *AIAA - 35th Aerospace Sciences Meeting & Exhibit*. No. Paper AIAA-97-0198.
- TERZOPOULOS, D. et METAXAS, D. (1991). Dynamic 3d models with local and global deformations: Deformable superquadrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. vol. 13, 703,714.
- YERRY, M. et SHEPHARD, M. (1984). Automatic three dimensional mesh generation by the modified octree technique. *International Journal for Numerical Methods in Engineering*, 20, 1965-1990.
- ZEEUW, D. D. (1993). *A Quadtree-Based Adaptively Refine Cartesian Grid Algorithm for Solution of the Euler Equations*. Thèse de doctorat, The University of Michigan.

Annexe A

Une présentation de MapleV

A.1 Généralités

MapleV est un produit de Waterloo Maple Software. Nous avons utilisé cet outil comme langage de programmation pour accélérer la phase de modélisation des objets et aussi lors de la phase de discrétisation. Il possède les caractéristiques suivantes:

- Fonctionne sous les quatres plateformes les plus importantes soit Windows 3.11, Windows 95, Windows 98, Windows NT et même Macintosh sans être trop exigeant sur la quantité de mémoire nécessaire.
- MapleV est un système dit ouvert en ce sens qu'il est possible de créer de nouvelles fonctionnalités selon notre besoin et de les incorporer ensuite dans le programme pour une utilisation ultérieure
- Contient un langage de programmation fonctionnel performant et flexible.

A.2 Modularité de MAPLE

Le logiciel mapleV est constitué de 4 modules principaux que l'on peut observer à la figure (A.1)

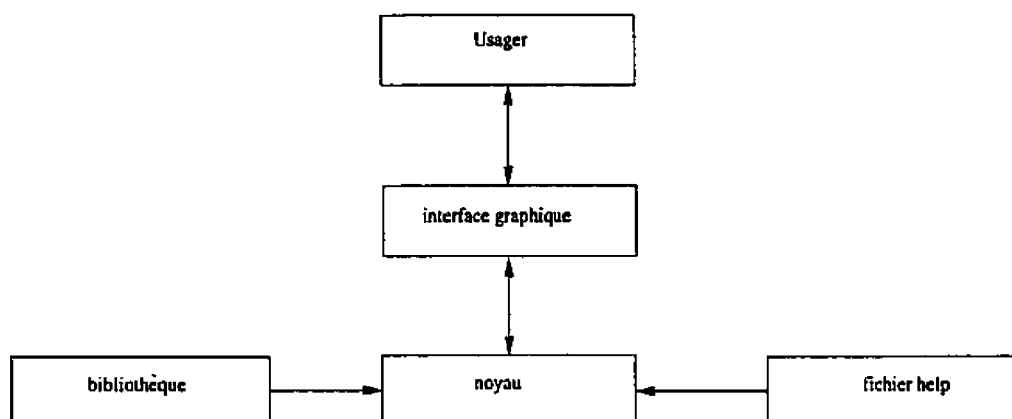


Figure A.1: Les quatres modules principaux de MapleV.

1. Le premier module s'occupe de l'interface graphique comme telle. Il est le lien direct avec l'usager et dépend de la version utilisée et de la plate-forme sur laquelle on travaille et est nommé Iris.
2. Le deuxième module voit à la vérification syntaxique, aux opérations algébriques de base, à la simplification, etc. On le nomme Kernel. Ces deux modules sont programmés essentiellement en C/C++ et sont chargés à chaque lancement du logiciel.
3. Le troisieme module est externe et est une Librairie. Il contient d'abord les bibliothèques spécialisées fournies par Waterloo Maple Software et aussi les contributions d'auteurs divers (shared library).
4. Le dernier module est constitué des fichiers d'aide en ligne qui sont des fichiers écrits dans le langage MapleV, ce qui en fait un système ouvert et facilement bonifiable par de nouvelles bibliothèques.

Les objets de bases que Maple manipule sont:

1. Les nombres qui peuvent être des entiers, des rationnels, des réels et des nombres complexes.
2. Une des forces de MapleV est qu'il n'est pas restreint qu'à des valeurs numériques. On peut aussi faire des opérations mathématiques sur des variables symboliques, avec la possibilité pour l'utilisateur de choisir n'importe

quel caractère ou chaîne de caractère. Toutefois il existe des caractères et chaînes de caractères qui sont utilisés par MapleV et qu'il est préférable de ne pas modifier.

3. Séquences, ensembles et listes Ce sont les trois objets privilégiés et fondamentaux de MapleV. Ces objets sont définis de la manière suivante:

- une séquence est une suite d'objets MapleV séparé par des virgules.
- un ensemble est une association non-ordonnée et sans répétition d'objets.
- une liste est une association ordonnée, avec répétition possible d'objets.

Une fois que le codage MapleV est terminé, on obtient une structure discrète de l'objet en appliquant deux routines de MapleV qui sont *grid* et *convert*, dans le but d'avoir une structure de type *polygons*. La routine *grid* = $[m, n]$ permet une discrétisation de l'objet et La routine *convert* traduit cette structure discrète (qui est en fait une liste de coordonnées) en une structure polygonale par utilisation de la macro commande *polygon*. Illustrons leur utilisation à travers le code MapleV ci-dessous:

```
with(plottools)
p:= plot3d(sin(x*y),x=-1..1,y=-1..1, grid=[2,2])
p1:=op(convert(p,POLYGONS))
display(p1,labels=[x,y,z])
```

A.3 Un exemple de fichier de type *polygon* généré par MapleV

Ce sont des fichiers de type structure de structures qui se présentent sous la forme: *polygon*($[[x_1, y_1], [x_2, y_2], \dots, [x_n, y_n]], \dots$) en 2D ou *polygon*($[[x_1, y_1, z_1], [x_2, y_2, z_2], \dots, [x_n, y_n, z_n]], \dots$) en 3D. L'exemple de fichier de type *polygon* que nous présentons ci-dessous est celui de *grid* = $[3, 3]$, qui génère la sous-figure (b) de la figure (A.3) et fournit la sortie ci-dessous. C'est ce type de

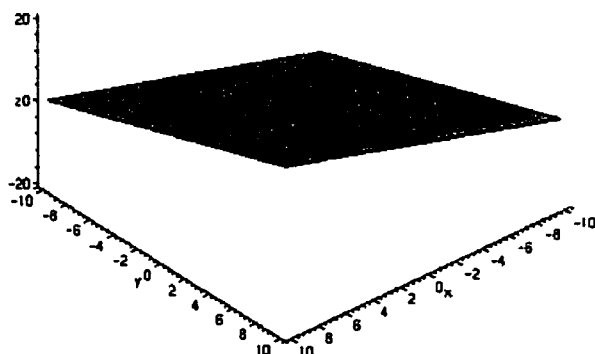


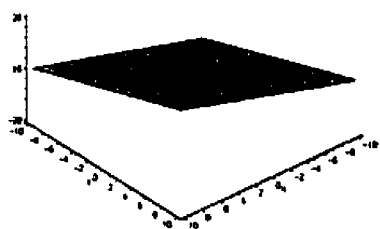
Figure A.2: Illustration graphique du code MapleV ci-dessus.

fichier issu de MapleV que l'on utilise pour exécuter le programme Octree tel que présenté dans la section (7.1.2) à travers les figures (7.1) et (7.2).

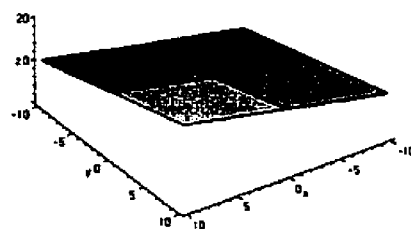
```
POLYGONS([[[-10, -10, -20], [-10, 0, -10], [0, 0, 0]],
           [[0, -10, -10], [0, 0, 0], [10, 0, 10], [10, -10, 0]],
           [[-10, 0, -10], [-10, 10, 0], [0, 10, 10], [0, 0, 0]],
           [[0, 0, 0], [0, 10, 10], [10, 10, 20], [10, 0, 10]]],
```

Le principe d'implémentation dans le noyau de MapleV des routines *polygon* et *grid* se présente sous la forme:

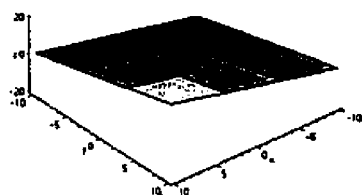
```
restart with(plottools)
interface(verboseproc=3)
eval(polygon)
proc(vlist)
local loptions, d: option
  if vlist <> [] then d := nops(op(1, vlist))fi:
  if d = 2 then loptions := convert(POLYGONS(evalf(vlist)
    op(loptions))):eval(display):
```



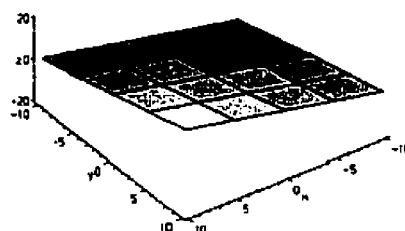
(a) Structure *polygon* de niveau 1.



(b) Structure *polygon* de niveau 2.



(c) Structure *polygon* de niveau 3.



(d) Structure *polygon* de niveau 4.

Figure A.3: Gradation de la discrétisation pour différentes valeurs de la macro commande *grid* sur un plan.



(a) Structure *polygon* de niveau 9.



(b) Structure *polygon* de niveau 19.



(c) Structure *polygon* de niveau 99.

Figure A.4: Gradation de la discrétisation pour différentes valeurs de la macro commande *grid* une sphère.

Annexe B

Mathématique des superquadriques

B.1 mathématique des superquadriques

Commençons par regarder la forme quadrique 2D la plus simple, c'est-à-dire le cercle trigonométrique, son équation paramétrique s'exprime sous la forme suivante:

$$\begin{aligned} x &= \cos \theta \\ y &= \sin \theta \quad -\pi \leq \theta < \pi. \end{aligned} \tag{B.1}$$

Pour alléger l'expression des calculs, nous utiliserons la notation vectorielle

$$\underline{p}(\theta) = \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} \quad -\pi \leq \theta < \pi. \tag{B.2}$$

Pour chacune des valeurs du paramètre θ , on obtient un vecteur \underline{p} qui détermine la position de la surface. en généralisant on a:

$$x = r_x |\cos(\theta)|^\epsilon \tag{B.3}$$

$$y = r_y |\sin(\theta)|^\epsilon \tag{B.4}$$

où θ est compris entre 0 et $\frac{\pi}{2}$ et r_x et r_y étant les rayons le long des axes x et y respectivement. La figure (B.1) montre donc une série de quart de superellipse pour différentes valeurs de ϵ . Pour obtenir des objets de taille différente, un poids est

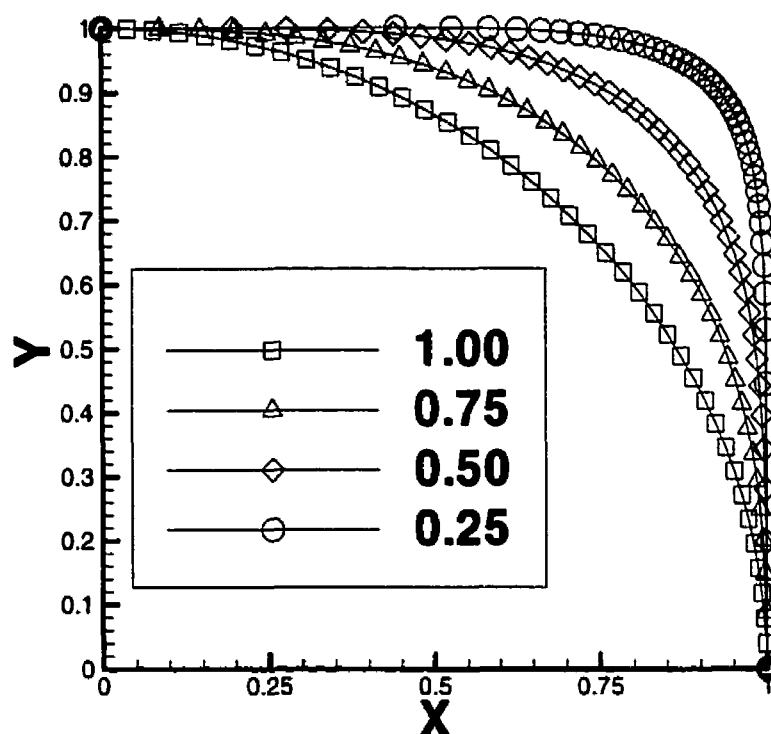


Figure B.1: Famille de quart de superellipse pour différentes valeurs de ϵ pour $r_x=r_y=1$.

ajouté aux fonctions trigonométriques (ce qui revient à comprimer ou allonger les axes). Il est ainsi possible d'obtenir une ellipse à partir d'un cercle. En utilisant d'autres fonctions trigonométriques, on obtient d'autres quadriques. L'hyperbole par exemple est obtenue en utilisant les fonctions trigonométriques sécante et tangente. Le tore, pour sa part, n'a pas d'équivalent en 2D. La valeur de l'exposant va influencer la courbure de la superquadrique. En modifiant l'exposant, on peut passer d'un objet rond à un autre dont la forme s'approche de celle d'un carré. Dans le cas de la superellipse, lorsque la valeur de ϵ est plus petite que 1, l'objet semble se gonfler pour se coller au parallélogramme défini par ses axes. Inversement, si ϵ devient

plus grand que 1, la surface de la superellipse va s'approcher de ses trois axes tout en restant liée à ceux-ci à ses extrémités. Les autres superquadriques réagissent d'une façon similaire en devenant plutôt carrée ou pointue selon la valeur de l'exposant. Il est intéressant de constater que quand l'exposant est 2, la quadrique devient plate. Le tableau suivant résume le comportement des superquadriques.

$\epsilon < 1$	L'objet devient carré
$\epsilon \sim 1$	L'objet est rond
$\epsilon \sim 2$	L'objet est plat
$\epsilon > 1$	L'objet devient pointu

Pour étendre les quadriques à l'espace 3D, on fait le produit sphérique de deux courbes 2D, noté $\underline{s} = \underline{c}_1 \otimes \underline{c}_2$. Si nous partons avec l'exemple d'une sphère, celle-ci peut s'exprimer comme étant le produit sphérique d'un cercle et d'un arc de cercle et on a:

$$\text{Demi-cercle : } \underline{c}_1 = \begin{bmatrix} \cos \eta \\ \sin \eta \end{bmatrix}, \quad -\pi/2 \leq \eta \leq \pi/2 \quad (\text{B.5})$$

$$\text{Cercle : } \underline{c}_2 = \begin{bmatrix} \cos \omega \\ \sin \omega \end{bmatrix}, \quad -\pi \leq \omega < \pi \quad (\text{B.6})$$

Leur produit sphérique nous donne la sphère

$$\underline{s} = \underline{c}_1 \otimes \underline{c}_2 = \begin{bmatrix} \cos \eta \cos \omega \\ \cos \eta \sin \omega \\ \sin \eta \end{bmatrix}, \quad \begin{array}{l} -\pi/2 \leq \eta \leq \pi/2 \\ -\pi \leq \omega < \pi. \end{array} \quad (\text{B.7})$$

Le demi-cercle module la taille du cercle selon son élévation en Z.

B.2 Autres composantes des superquadriques

Les superquadriques sont des solides aux surfaces stables. Ceci permet de dériver des fonctions utiles qui seront elles aussi stables. L'équation du vecteur normal à la surface pour une ellipse s'obtient par la formule suivante:

$$\underline{n}(\eta, \omega) = \begin{bmatrix} 1/a_1 \cos_{\eta}^{2-\epsilon_1} \cos_{\omega}^{2-\epsilon_2} \\ 1/a_2 \cos_{\eta}^{2-\epsilon_1} \sin_{\omega}^{2-\epsilon_2} \\ 1/a_3 \sin_{\eta}^{2-\epsilon_1} \end{bmatrix}, \quad \begin{array}{l} -\pi/2 \leq \eta \leq \pi/2 \\ -\pi \leq \omega < \pi. \end{array} \quad (\text{B.8})$$

Le vecteur normal à la surface est équivalent au vecteur de surface d'un superquadrique dual, c.à.d celui dont les poids des fonctions trigonométriques sont inversés et dont le nouvel exposant est égal à $2 - \epsilon$ pour un exposant initial de ϵ . Ceci est facile à constater sur le cercle ou la sphère, car $\epsilon = 2 - \epsilon = 1$. Les superquadriques possèdent une fonction ϕ , intérieur/extérieur stable qui donne la position d'un point par rapport au superquadrique. Cette équation est simple et est dérivée de la forme implicite de l'équation du superquadrique. La position dépend de la valeur de la fonction, et nous avons les cas de figure ci-dessous:

$$\text{Si } f(x, y, z) \begin{cases} < 1, & (x, y, z) \text{ est dans le solide} \\ = 1, & (x, y, z) \text{ est sur la surface} \\ > 1, & (x, y, z) \text{ est hors du solide} \end{cases} \quad (\text{B.9})$$

Dans le cas de l'hyperbole, un point est considéré à l'extérieur s'il se trouve dans la partie concave de l'hyperbole.

B.3 Quelques formules mathématiques de base des superquadriques

Barr (1981) a effectué à cet sujet une étude détaillée. La superellipse s'obtient du produit sphérique de deux courbes 2D, soit une demi-superellipse et une superellipse

$$\underline{p} = \begin{bmatrix} \cos_{\eta}^{\epsilon_1} \\ a_3 \sin_{\eta}^{\epsilon_1} \end{bmatrix} \otimes \begin{bmatrix} a_1 \cos_{\omega}^{\epsilon_2} \\ a_2 \sin_{\omega}^{\epsilon_2} \end{bmatrix} \quad \begin{array}{l} -\pi/2 \leq \eta \leq \pi/2 \\ -\pi \leq \omega < \pi. \end{array} \quad (\text{B.10})$$

D'où l'équation paramétrique des superellipses:

$$\underline{p}(\eta, \omega) = \begin{bmatrix} a_1 \cos_{\eta}^{\epsilon_1} \cos_{\omega}^{\epsilon_2} \\ a_2 \cos_{\eta}^{\epsilon_1} \sin_{\omega}^{\epsilon_2} \\ a_3 \sin_{\eta}^{\epsilon_1} \end{bmatrix}, \quad \begin{array}{l} -\pi/2 \leq \eta \leq \pi/2 \\ -\pi \leq \omega < \pi. \end{array} \quad (\text{B.11})$$

Les équations implicite et explicite nous donnent respectivement:

$$((x/a_1)^{2/\epsilon_2} + (y/a_2)^{2/\epsilon_2})^{\epsilon_2/\epsilon_1} + (z/a_3)^{2/\epsilon_1} = 1 \quad (\text{B.12})$$

et

$$z = \pm a_3 (1 - ((x/a_1)^{2/\epsilon_2} + (y/a_2)^{2/\epsilon_2})^{\epsilon_2/\epsilon_1})^{\epsilon_1/2}. \quad (\text{B.13})$$

Le vecteur normal est donnée par:

$$\underline{n}(\eta, \omega) = \begin{bmatrix} 1/a_1 \cos_{\eta}^{2-\epsilon_1} \cos_{\omega}^{2-\epsilon_2} \\ 1/a_2 \cos_{\eta}^{2-\epsilon_1} \sin_{\omega}^{2-\epsilon_2} \\ 1/a_3 \sin_{\eta}^{2-\epsilon_1} \end{bmatrix}, \quad \begin{array}{l} -\pi/2 \leq \eta \leq \pi/2 \\ -\pi \leq \omega < \pi \end{array} \quad (\text{B.14})$$

et la fonction ϕ , intérieur/extérieur est

$$f(x, y, z) = ((x/a_1)^{2/\epsilon_2} + (y/a_2)^{2/\epsilon_2})^{\epsilon_2/\epsilon_1} + (z/a_3)^{2/\epsilon_1} \quad (\text{B.15})$$

B.4 Superhyperboles

La superhyperbole simple s'obtient du produit sphérique de deux courbes 2D, soit une superhyperbole et une superellipse.

$$\underline{p} = \begin{bmatrix} \text{Sec}_\eta^{\epsilon_1} \\ a_3 \text{Tan}_\eta^{\epsilon_1} \end{bmatrix} \otimes \begin{bmatrix} a_1 \text{Cos}_\omega^{\epsilon_2} \\ a_2 \text{Sin}_\omega^{\epsilon_2} \end{bmatrix} \quad \begin{array}{l} -\pi/2 < \eta < \pi/2 \\ -\pi \leq \omega < \pi \end{array} \quad (\text{B.16})$$

D'où l'équation paramétrique des superhyperboles simples

$$\underline{p}(\eta, \omega) = \begin{bmatrix} a_1 \text{Sec}_\eta^{\epsilon_1} \text{Cos}_\omega^{\epsilon_2} \\ a_2 \text{Sec}_\eta^{\epsilon_1} \text{Sin}_\omega^{\epsilon_2} \\ a_3 \text{Tan}_\eta^{\epsilon_1} \end{bmatrix}, \quad \begin{array}{l} -\pi/2 < \eta < \pi/2 \\ -\pi \leq \omega < \pi. \end{array} \quad (\text{B.17})$$

Les équations implicite et explicite nous donnent

$$((x/a_1)^{2/\epsilon_2} + (y/a_2)^{2/\epsilon_2})^{\epsilon_2/\epsilon_1} - (z/a_3)^{2/\epsilon_1} = 1$$

et

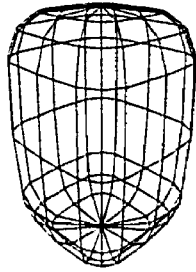
$$z = \pm a_3 ((x/a_1)^{2/\epsilon_2} + (y/a_2)^{2/\epsilon_2})^{\epsilon_2/\epsilon_1} - 1)^{\epsilon_1/2}$$

Le vecteur normal est donné par:

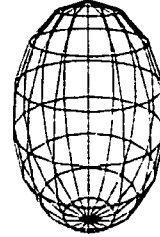
$$\underline{n}(\eta, \omega) = \begin{bmatrix} 1/a_1 \text{Sec}_\eta^{2-\epsilon_1} \text{Cos}_\omega^{2-\epsilon_2} \\ 1/a_2 \text{Sec}_\eta^{2-\epsilon_1} \text{Sin}_\omega^{2-\epsilon_2} \\ 1/a_3 \text{Tan}_\eta^{2-\epsilon_1} \end{bmatrix}, \quad \begin{array}{l} -\pi/2 < \eta < \pi/2 \\ -\pi \leq \omega < \pi \end{array} \quad (\text{B.18})$$

et la fonction ϕ , intérieur/extérieur par:

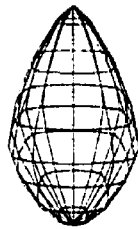
$$f(x, y, z) = ((x/a_1)^{2/\epsilon_2} + (y/a_2)^{2/\epsilon_2})^{\epsilon_2/\epsilon_1} - (z/a_3)^{2/\epsilon_1}. \quad (\text{B.19})$$



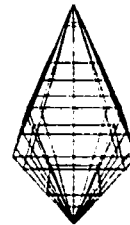
(a) Superellipse $\epsilon_{1,2} = 0.5$



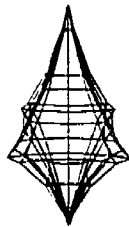
(b) Superellipse pour $\epsilon_{1,2} = .10$



(c) Superellipse pour $\epsilon_{1,2} = .15$



(d) Superellipse pour $\epsilon_{1,2} = .20$



(e) Superellipse pour $\epsilon_{1,2} = .25$

Figure B.2: Superellipses avec différentes valeurs pour ϵ_1 et ϵ_2

B.5 Superhyperboles doubles

La superhyperbole double s'obtient du produit sphérique de deux courbes 2D, soit une superhyperbole et une seconde prise cette fois-ci sur deux périodes de longueur π .

$$\underline{p} = \begin{bmatrix} \text{Sec}_\eta^{\epsilon_1} \\ a_3 \text{Tan}_\eta^{\epsilon_1} \end{bmatrix} \otimes \begin{bmatrix} a_1 \text{Sec}_\omega^{\epsilon_2} \\ a_2 \text{Tan}_\omega^{\epsilon_2} \end{bmatrix} \quad \begin{array}{l} -\pi/2 < \eta < \pi/2 \\ -\pi/2 < \omega < \pi/2 \\ \pi/2 < \omega < 3\pi/2 \end{array} \quad (\text{B.20})$$

D'où l'équation paramétrique des superhyperboles doubles:

$$\underline{p}(\eta, \omega) = \begin{bmatrix} a_1 \text{Sec}_\eta^{\epsilon_1} \text{Sec}_\omega^{\epsilon_2} \\ a_2 \text{Sec}_\eta^{\epsilon_1} \text{Tan}_\omega^{\epsilon_2} \\ a_3 \text{Tan}_\eta^{\epsilon_1} \end{bmatrix}, \quad \begin{array}{l} -\pi/2 < \eta < \pi/2 \\ -\pi/2 < \omega < \pi/2 \\ \pi/2 < \omega < 3\pi/2. \end{array} \quad (\text{B.21})$$

Les équations implicite et explicite nous donnent:

$$((x/a_1)^{2/\epsilon_2} - (y/a_2)^{2/\epsilon_2})^{\epsilon_2/\epsilon_1} - (z/a_3)^{2/\epsilon_1} = 1 \quad (\text{B.22})$$

et

$$z = \pm a_3 ((x/a_1)^{2/\epsilon_2} - (y/a_2)^{2/\epsilon_2})^{\epsilon_2/\epsilon_1} - 1)^{\epsilon_1/2}. \quad (\text{B.23})$$

Le vecteur normal est donné par:

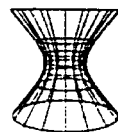
$$\underline{n}(\eta, \omega) = \begin{bmatrix} 1/a_1 \text{Sec}_\eta^{2-\epsilon_1} \text{Sec}_\omega^{2-\epsilon_2} \\ 1/a_2 \text{Sec}_\eta^{2-\epsilon_1} \text{Tan}_\omega^{2-\epsilon_2} \\ 1/a_3 \text{Tan}_\eta^{2-\epsilon_1} \end{bmatrix}, \quad \begin{array}{l} -\pi/2 < \eta < \pi/2 \\ -\pi/2 < \omega < \pi/2 \\ \pi/2 < \omega < 3\pi/2 \end{array} \quad (\text{B.24})$$

et la fonction ϕ , intérieur/extérieur par:

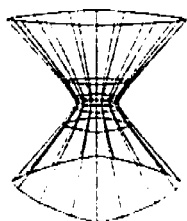
$$f(x, y, z) = ((x/a_1)^{2/\epsilon_2} - (y/a_2)^{2/\epsilon_2})^{\epsilon_2/\epsilon_1} - (z/a_3)^{2/\epsilon_1}. \quad (\text{B.25})$$



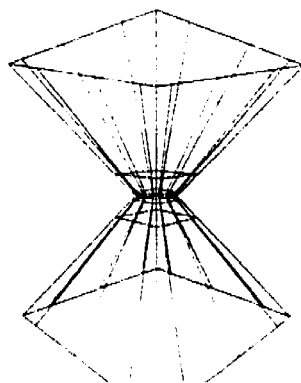
(a) Superhyperbole pour $\epsilon = 0.5$



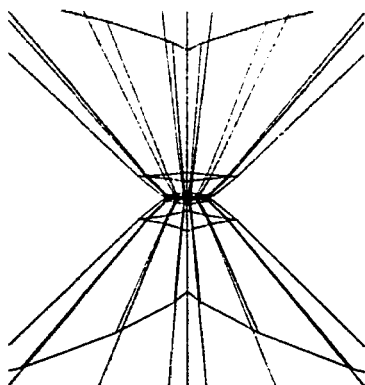
(b) Superhyperbole pour $\epsilon = .10$



(c) Superhyperbole pour $\epsilon = .15$



(d) Superhyperbole pour $\epsilon = .20$



(e) Superhyperbole pour $\epsilon = .25$

Figure B.3: Superhyperboles simples avec différentes valeurs pour ϵ_1 et ϵ_2 .

B.6 Supertore

Le supertore s'obtient du produit sphérique de deux courbes 2D, soit une superellipse translaturée en X et une superellipse.

$$\underline{p} = \begin{bmatrix} a_4 + \text{Cos}_\eta^{\epsilon_1} \\ a_3 \text{Sin}_\eta^{\epsilon_1} \end{bmatrix} \otimes \begin{bmatrix} a_1 \text{Cos}_\omega^{\epsilon_2} \\ a_2 \text{Sin}_\omega^{\epsilon_2} \end{bmatrix} \quad \begin{matrix} -\pi \leq \eta < \pi \\ -\pi \leq \omega < \pi \end{matrix} \quad (\text{B.26})$$

où $a_4 = \hat{a} / \sqrt{a_1^2 + a_2^2}$ et \hat{a} est le rayon du tore. De là, l'équation paramétrique des supertores est

$$\underline{p}(\eta, \omega) = \begin{bmatrix} a_1(a_4 + \text{Cos}_\eta^{\epsilon_1}) \text{Cos}_\omega^{\epsilon_2} \\ a_2(a_4 + \text{Cos}_\eta^{\epsilon_1}) \text{Sin}_\omega^{\epsilon_2} \\ a_3 \text{Sin}_\eta^{\epsilon_1} \end{bmatrix}, \quad \begin{matrix} -\pi \leq \eta < \pi \\ -\pi \leq \omega < \pi. \end{matrix} \quad (\text{B.27})$$

Les équations implicite et explicite nous donnent

$$(((x/a_1)^{2/\epsilon_2} + (y/a_2)^{2/\epsilon_2})^{\epsilon_2/2} - a_4)^{2/\epsilon_1} + (z/a_3)^{2/\epsilon_1} = 1 \quad (\text{B.28})$$

et

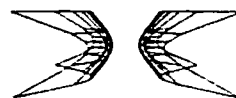
$$z = \pm a_3 (1 - (((x/a_1)^{2/\epsilon_2} + (y/a_2)^{2/\epsilon_2})^{\epsilon_2/2} - a_4)^{2/\epsilon_1})^{\epsilon_1/2}. \quad (\text{B.29})$$

Le vecteur normal est donné par:

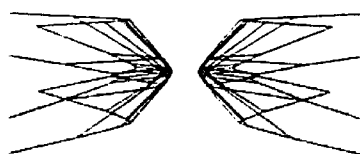
$$\underline{n}(\eta, \omega) = \begin{bmatrix} 1/a_1 \text{Cos}_\eta^{2-\epsilon_1} \text{Cos}_\omega^{2-\epsilon_2} \\ 1/a_2 \text{Cos}_\eta^{2-\epsilon_1} \text{Sin}_\omega^{2-\epsilon_2} \\ 1/a_3 \text{Sin}_\eta^{2-\epsilon_1} \end{bmatrix}, \quad \begin{matrix} -\pi \leq \eta < \pi \\ -\pi \leq \omega < \pi \end{matrix} \quad (\text{B.30})$$



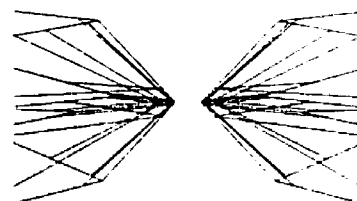
(a) Superhyperbole double pour $\epsilon_{1,2} = 0.5$



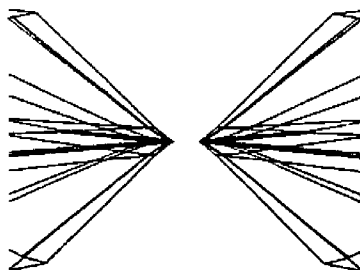
(b) Superhyperbole double pour $\epsilon_{1,2} = .10$



(c) Superhyperbole double pour $\epsilon_{1,2} = .15$

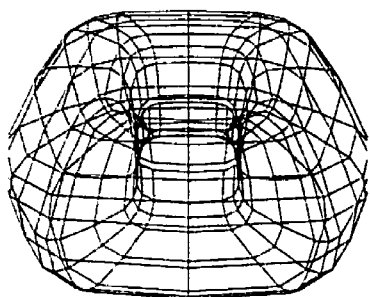


(d) Superhyperbole double pour $\epsilon_{1,2} = .20$

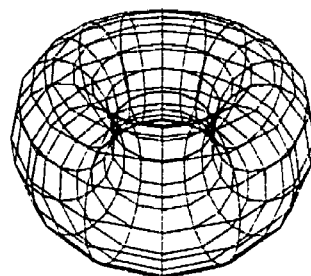


(e) Superhyperbole double pour $\epsilon_{1,2} = .25$

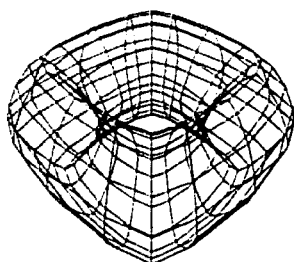
Figure B.4: Superhyperboles doubles avec différentes valeurs pour ϵ_1 et ϵ_2 .



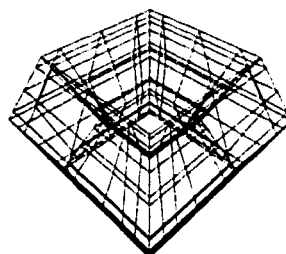
(a) Supertore pour $\epsilon_{1,2} = 0.5$



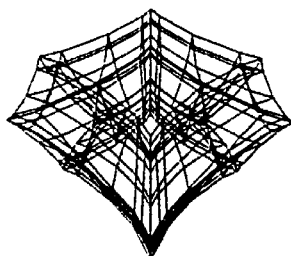
(b) Supertore pour $\epsilon_{1,2} = .10$



(c) Supertore pour $\epsilon_{1,2} = .15$



(d) Supertore pour $\epsilon_{1,2} = .20$



(e) Supertore pour $\epsilon_{1,2} = .25$

Figure B.5: Supertores avec différentes valeurs pour ϵ_1 et ϵ_2 .

B.7 Superquadriques augmentés de transformations

Lors de l'utilisation de primitives dans les différentes applications, des transformations devront être utilisées. Les superquadriques ont un bon comportement mathématique sous les transformations de sorte que le vecteur de surface et la fonction ϕ , intérieur/extérieur se calculent relativement facilement. Le vecteur normal quant à lui pourrait demander plus de calculs.

Pour des transformations de rotation et de translation, les équations qui suivent décrivent le comportement du vecteur de surface \underline{v} , du vecteur normal \underline{n} et de la fonction ϕ , intérieur/extérieur f . Soit:

M	la matrice de rotation	
M^{-1}	la matrice de rotation inverse	
\underline{b}	le vecteur de translation	(B.31)
$\hat{\underline{v}}, \hat{\underline{n}}$	les vecteurs transformés	
\hat{f}	la fonction ϕ , intérieur/extérieur pour les points transformés	

Nous avons alors:

$$\begin{aligned}\hat{\underline{v}} &= M \underline{v} + \underline{b} \\ \hat{\underline{n}} &= M \underline{n} \\ \hat{f}(\hat{x}, \hat{y}, \hat{z}) &= f(x, y, z)\end{aligned}\tag{B.32}$$

où

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = M^{-1} \begin{bmatrix} \hat{x} - b_x \\ \hat{y} - b_y \\ \hat{z} - b_z \end{bmatrix}\tag{B.33}$$

Il faut ici noter que la matrice M^{-1} est égale à M^T , la transposée de M . Ceci est vrai à cause de l'orthogonalité de la matrice de rotation. Si nous considérons une

transformation quelconque \mathcal{T} , les équations deviennent plutôt

$$\underline{\hat{v}} = \mathcal{T}(\underline{v}) \quad (\text{B.34})$$

et

$$\hat{f}(\hat{x}, \hat{y}, \hat{z}) = f\left(\mathcal{T}^{-1}\left(\begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{bmatrix}\right)\right). \quad (\text{B.35})$$

B.7.1 Transformations globales

Outre les transformations de rotation et translation, les transformations globales introduites dans Solina et Bajcsy (1990) permettent de définir des objets très utiles par leurs formes se rapprochant d'objets susceptibles de se retrouver en industrie. Plusieurs transformations peuvent être imaginées, mais les plus utiles sont l'effilement (tapering), la courbure (bending), et la cavité (cavity). On introduira les transformations en considérant qu'elles seront appliquées à une superellipse centrée à l'origine. Les transformations seront définies par rapport à l'axe Z . Comme nous l'avons vu à la section (B.34), il est utile de faire la transformation inverse des coordonnées pour retrouver la fonction ϕ , intérieur/extérieur. Nous définirons donc les transformations en fonction de la conversion des coordonnées transformées X , Y et Z vers les coordonnées non transformées x , y , z .

B.7.2 Courbure

La courbure est, comme l'effilement, définie par rapport à l'axe Z . L'angle α dans le plan XY déterminera le plan de courbure P_c et la direction de courbure D_c . Tous les points de l'objet seront plus ou moins courbés selon la valeur de leur composante Z . Ils se courberont dans la direction D_c et parallèlement au plan P_c . Par exemple, un segment partant de l'origine et suivant l'axe Z deviendra un arc de cercle dans le plan de courbure. Le paramètre k^{-1} fournit le rayon de courbure. L'objet s'enroulera donc autour d'une droite dans XY perpendiculaire à P_c et à une distance k^{-1} de l'origine. Pour un point (x, y, z) à courber, il faut tout d'abord

trouver l'angle β qu'il fait dans le plan XY.

$$\beta = \arctan \frac{x}{y} \quad (\text{B.36})$$

De là, le point doit être projeté sur le plan P_c . Nous ignorons ici la coordonnée Z du point et ne voulons retrouver que la composante parallèle à XY. Le scalaire r donnera la projection du point (x, y) sur la droite formée par les plans XY et P_c

$$r = \cos(\alpha - \beta) \sqrt{x^2 + y^2}. \quad (\text{B.37})$$

L'angle de courbure est définit par:

$$\gamma = zk^{-1}. \quad (\text{B.38})$$

La courbure va transformer r en

$$R = k^{-1} - \cos(\gamma)(k^{-1} - r). \quad (\text{B.39})$$

Le point transformé est:

$$X = x + \cos(\alpha)(R - r)$$

$$Y = y + \sin(\alpha)(R - r) \quad (\text{B.40})$$

$$Z = \sin(\gamma)(k^{-1} - r).$$

De là, la transformation inverse:

$$x = X - \cos(\alpha)(R - r)$$

$$y = Y - \sin(\alpha)(R - r) \quad (\text{B.41})$$

$$z = k^{-1}\gamma$$

Pour

$$\gamma = \arctan \frac{Z}{k^{-1} - R}$$

$$r = k^{-1} - \sqrt{Z^2 + (k^{-1} - R)^2} \quad (\text{B.42})$$

$$R = \cos(\alpha - \arctan \frac{Y}{X}) \sqrt{X^2 + Y^2}.$$

Cette formulation a un inconvénient majeur du fait que k ne peut être zéro pour représenter une courbure nulle. Il est cependant possible d'initialiser k à une très petite valeur lors de l'implantation. Ceci nous donnera un rayon de courbure quasi infini. La transformation de courbure se définit par les paramètres k et α .

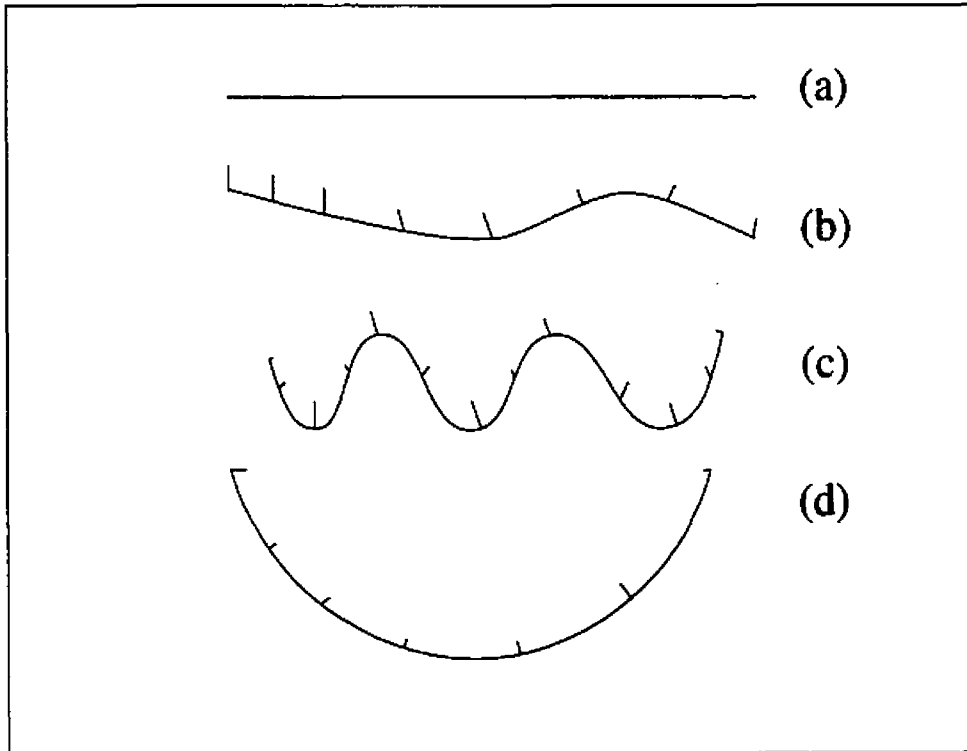


Figure B.6: Illustration en 2D de la notion de courbure pour différentes valeurs de α et k . On montre l'existence de quelques points de discontinuité si la courbure est trop forte.

B.7.3 Cavité

La dernière transformation qui sera introduite est la cavité. Comme toutes les autres, elle se définit par rapport à l'axe Z. La cavité est le fait de recourber vers le haut un objet selon un certain angle et à une certaine distance de l'axe Z. Pour faire une déformation de cavité à un point (x, y, z) , les coordonnées x et y de ce dernier sont tout d'abord transformées en coordonnées polaires.

$$r = \sqrt{x^2 + y^2} \quad (\text{B.43})$$

$$\alpha = \arctan \frac{y}{x}$$

Comme pour la courbure, la cavité a un rayon k^{-1} déterminant la courbure de la cavité. La cavité possède aussi un intervalle de courbure $r_0..r_1$. Pour définir l'angle de courbure, nous avons besoin de définir

$$\hat{r} = \begin{cases} r_0 & \text{si } r < r_0 \\ r & \text{si } r_0 < r < r_1 \\ r_1 & \text{si } r_1 < r \end{cases} \quad (\text{B.44})$$

Et l'angle de courbure:

$$\gamma = k(\hat{r} - r_0) \quad (\text{B.45})$$

est défini de sorte qu'il soit nul avant r_0 , augmente de r_0 à r_1 et reste constant après r_1 . L'objet restera inchangé pour tous les points dont $r \leq r_0$. Aux points pour lesquels $r_0 < r < r_1$, une certaine courbure sera appliquée en retroussant de plus en plus les points vers le haut (si $k > 0$) autour de l'axe Z. Quand $r_1 \leq r$, les points de l'objet sont recourbés vers le haut autour de l'axe Z avec un angle constant. Il est maintenant possible de définir R , la composante transformée de r

$$R = \begin{cases} r & \text{si } r < r_0 \\ r_0 + \sin(\gamma)(k^{-1} - z) & \text{si } r_0 < r < r_1 \\ r_0 + \sin(\gamma)(k^{-1} - z) + \cos(\gamma)(r - r_1) & \text{si } r_1 < r \end{cases} \quad (\text{B.46})$$

En définitive le point transformé est:

$$\begin{aligned} X &= R \cos(\alpha) \\ Y &= R \sin(\alpha) \end{aligned} \tag{B.47}$$

$$Z = \begin{cases} z & \text{si } r < r_0 \\ k^{-1} - \cos(\gamma)(k^{-1} - z) & \text{si } r_0 < r < r_1 \\ k^{-1} - \cos(\gamma)(k^{-1} - z) \sin(\gamma)(r - r_1) & \text{si } r_1 < r \end{cases}$$

La transformation inverse sera définie comme suit

$$R = \sqrt{X^2 + Y^2} \tag{B.48}$$

Si $R < r_0$ alors

$$\begin{aligned} x &= X \\ y &= Y \\ z &= Z. \end{aligned} \tag{B.49}$$

Si $r_0 < R$, alors on doit calculer

$$\begin{aligned} \alpha &= \arctan \frac{Y}{X} \\ \gamma &= \arctan \frac{R - r_0}{k^{-1} - Z} \\ r &= r_0 + \frac{\gamma}{k} \end{aligned} \tag{B.50}$$

De là, si $r < r_1$ alors

$$\begin{aligned}
 x &= r \sin(\alpha) \\
 y &= r \cos(\alpha) \\
 z &= \frac{Z - k^{-1}}{\cos \gamma} + k^{-1}
 \end{aligned}
 \tag{B.51}$$

Si $r_1 \leq r$ alors on recalcule:

$$\begin{aligned}
 \gamma &= k(r_1 - r_0) \\
 r &= r_1 + \cos(\gamma)(R - r_0) + \sin(\gamma)(Z - k^{-1})
 \end{aligned}
 \tag{B.52}$$

et nous avons en définitive:

$$\begin{aligned}
 x &= r \sin(\alpha) \\
 y &= r \cos(\alpha) \\
 z &= \frac{1}{\cos \gamma}(Z - k^{-1} - \sin(\gamma)(r - r_1) + \frac{\cos \gamma}{k})
 \end{aligned}
 \tag{B.53}$$

La transformation de cavité est définie par les paramètres r_0 , r_1 et k .

B.7.4 Transformations locales

Des transformations locales peuvent être ajoutées aux transformations globales pour mieux modéliser certaines parties de l'objet qui ne sont pas représentées par le modèle global. Dans Terzopoulos et Metaxas (1991), l'utilisation de membrane spline, permet de modéliser des objets aux contours plutôt irréguliers. Les transformations locales rendent le modèle moins utilisable pour la reconnaissance de forme car des parties cachées de l'objet peuvent contenir des déformations locales.